



UNIVERSIDAD CARLOS III DE MADRID

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

PROYECTO FIN DE CARRERA

***DISEÑO DEL SISTEMA DE GUIADO DE UN AUTOMÓVIL
AUTÓNOMO***

AUTOR: Gema Morales Gavilán

TUTOR: Fernando García Fernández

A mi madre

Agradecimientos

Ha sido un largo camino hasta alcanzar este momento...son muchas las cosas que agradecer, y tantas las personas que han caminado a mi lado, que resulta extremadamente complicado poder plasmarlo en palabras...

A mi familia, por ayudarme a construir lo que soy; a mi padre, hermanos y abuelas, por su apoyo incondicional, porque siempre han confiado en mí y en mis capacidades, y por darme la fuerza para levantarme una y otra vez, incluso tras las caídas más duras...sois lo más importante para mí; a las Mari Luz, a ambas, por alumbrarme y dar color al mundo; a mis tíos y primos, y a toda mi familia en general, gracias.

A mis niñas: Luci, Cris, Noe y Marga, por tantos años y tantos buenos momentos vividos. No sé qué hubiera hecho sin vosotras, os quiero.

A la resi, mi resi, que fue mi hogar y refugio; a todos los que conocí allí: Merino, Guille, Pablo, Dani, Carlos...a Pepi por su cariño; a todos, muchas gracias.

A esos linenses que han estado tan lejos en la distancia, pero tan cerca en el corazón: a Melani, por ser más que una amiga, una hermana; a Guti, por su paciencia infinita y esas visitas a Gibraltar; a Fifi, por preocuparse siempre de cómo estaba; a Yayo, por hacerme reír y enseñarme que la vida es tan buena como queramos que sea; a Toni y Canela, porque los pelos largos y las chupas de cuero esconden corazones enormes; a Alberto, Vero, Joe, Juan, y a todos y cada uno de vosotros, gracias.

A los que estuvisteis conmigo todos estos años de carrera, por no haber sido sólo compañeros, sino por encima de todo, amigos: Gema – por esa fuerza y esa amistad sin límites- , Nacho – qué grande eres, y cuánto me has aguantado- , Vito, Guille, Eli, Teresa, María, Pilar, Kike, Fran, Sergio, el otro Sergio, Pedrito...sois muchos y aunque no ponga todos los nombres, sabéis quienes sois...a todos, gracias.

A los que he tenido el inmenso placer de conocer en este tiempo, y que tanto me han ayudado y a los que tanto he conseguido (a veces) sacar de quicio: a Saúl, por

ser el primero en ofrecer una mano amiga (y el primero en hacerme rabiar); a Harold, por su confianza y amistad; a Carlos, por sus consejos y por disfrutar tanto con su compañía; a Gorka, por su paciencia ante mis neuras...a todos los que me dejó en el tintero, pero siguen mi mente, a todos, gracias.

A la gente del departamento, que me han hecho sentir parte de una gran familia: a José María, Arturo y Paco Pepe, por tratarme con tanto afecto y darme la oportunidad de estar donde estoy; a María, por ser como es y porque sin ella nada hubiera sido tan bueno; a Basam y Dani, por su apoyo incondicional y ese humor sarcástico que me encanta; a Marco, por ser atento y dulce; a Gustavo, por la felicidad que irradia a su alrededor; a Juampa, porque aunque no esté aquí, no le olvidamos; a los que han estado conmigo en esta aventura, y que tantas horas de trabajo y risas hemos pasado...y por supuesto a Nando, por aguantarme durante todo este tiempo, por ser mi tutor, mi compañero, mi consejero, mi psicólogo, y por encima de todo, un gran amigo.

Aunque lo haya dejado para el final, el primero de los agradecimientos es para ella: para mi madre, con todo mi cuerpo y alma, con todo mi ser...ojalá estuvieses aquí, porque sé que, a pesar de mis fallos, te sentirías muy orgullosa. Soy lo que soy por ti.

A todos y cada uno de vosotros...muchas, muchas gracias.

Índice general

AGRADECIMIENTOS.....	V
ÍNDICE GENERAL.....	VII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS.....	XV
CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1 NECESIDADES A CUBRIR EN EL PROYECTO.....	2
1.2 OBJETIVOS A DESARROLLAR.....	4
1.3 FASES DEL PROYECTO	5
1.4 PROBLEMAS EN LA PLANIFICACIÓN DE CAMINOS	7
1.5 ESTRUCTURA DEL DOCUMENTO	9
CAPÍTULO 2. ESTADO DEL ARTE.....	11
2.1 INTRODUCCIÓN AL CONCEPTO DE ROBOTS MÓVILES AUTÓNOMOS	12
2.2 MÉTODOS CLÁSICOS DE PLANIFICACIÓN	14
2.3 CONSTRUCCIÓN DE MAPAS.....	15
2.3.1 Mapas métricos.....	15
2.3.2 Mapas topológicos	17
2.3.3 Mapas semánticos	18
2.4 MAPAS DE CARRETERA (ROADMAPS)	19
2.4.1 Grafo de visibilidad.....	19
2.4.2 Diagrama de Voronoi.....	21
2.4.3 Caminos probabilísticos (PRM)	22

2.5	DESCOMPOSICIÓN EN CELDAS	23
2.6	CAMPOS DE POTENCIAL.....	26
CAPÍTULO 3. ENTORNO DE DESARROLLO.....		31
3.1	ARQUITECTURA HARDWARE.....	32
3.1.1	<i>Telémetro láser</i>	<i>34</i>
3.1.2	<i>Adquisición y Procesado</i>	<i>35</i>
3.1.3	<i>Alimentación.....</i>	<i>36</i>
3.1.4	<i>Visualización y E/S.....</i>	<i>37</i>
3.2	ARQUITECTURA SOFTWARE	37
3.3	SISTEMAS IMPLEMENTADOS O EN DESARROLLO	40
CAPÍTULO 4. PERCEPCIÓN DEL ENTORNO FÍSICO		47
4.1	PARTICULARIZACIÓN DEL ENTORNO.....	49
4.2	SISTEMA DE PERCEPCIÓN.....	50
4.2.1	<i>Fase de adquisición - dispositivo láser</i>	<i>51</i>
4.2.2	<i>Fase de tratamiento - clasificación de la información</i>	<i>57</i>
4.2.3	<i>Fase de interpretación - modelado de obstáculos</i>	<i>57</i>
CAPÍTULO 5. LOCALIZACIÓN EN EL ENTORNO FÍSICO.....		61
5.1	SISTEMA DE LOCALIZACIÓN.....	62
5.2	TECNOLOGÍA GPS	64
CAPÍTULO 6. ESQUEMA DE PROGRAMACIÓN.....		71
6.1	LIBRERÍA PARA IMPLEMENTACIONES EN ROBÓTICA: MRPT	72
6.2	LIBRERÍAS DE SOPORTE	77

6.3	CLASES PRINCIPALES EMPLEADAS EN EL DESARROLLO DEL SISTEMA.....	78
-----	--	----

CAPÍTULO 7. IMPLEMENTACIÓN.....81

7.1	MAPEADO MEDIANTE CELDAS DE OCUPACIÓN	82
7.1.1	<i>Base teórica</i>	82
7.1.2	<i>Aprendizaje inverso en los modelos de medida.....</i>	84
7.1.3	<i>Criterio MAP (Maximun A Posterior Occupancy Grid Mapping).....</i>	85
7.2	INTEGRACIÓN DE COMPONENTES AL PROYECTO	87
7.3	ALGORITMOS DE BÚSQUEDA GLOBAL.....	88
7.3.1	<i>Mapeado global del entorno mediante celdas de ocupación.....</i>	88
7.3.2	<i>Planificación global.....</i>	91
7.4	ALGORITMOS DE BÚLSQUEDA LOCAL	95
7.4.1	<i>Mapeado local del entorno mediante celdas de ocupación.....</i>	95
7.4.2	<i>Planificación local.....</i>	105
7.5	MAPEADO INCREMENTAL MEDIANTE EL MODELO DE OBSTÁCULOS	113
7.5.1	<i>Incorporación de los obstáculos</i>	113

CAPÍTULO 8. RESULTADOS.....123

8.1	PRUEBAS EN ESTÁTICO	124
8.1.1	<i>Simulación 1.....</i>	125
8.1.2	<i>Simulación 2.....</i>	126
8.1.3	<i>Simulación 3.....</i>	130
8.1.4	<i>Simulación 4.....</i>	132
8.2	PRUEBAS EN MOVIMIENTO	136
8.2.1	<i>Simulación 1.....</i>	136
8.2.2	<i>Simulación 2.....</i>	139
8.2.3	<i>Simulación 3.....</i>	142

8.2.4 Simulación 4.....	144
8.2.5 Simulación 5.....	147
8.3 COMPARATIVA DE PRUEBAS.....	150
CAPÍTULO 9. CONCLUSIONES Y DESARROLLOS FUTUROS	155
9.1 CONCLUSIONES	156
9.2 DESARROLLOS FUTUROS.....	158
BIBLIOGRAFÍA	161
ANEXOS	165
TRANSFORMACIONES DE COORDENADAS	166

Índice de figuras

FIGURA 1-1: ESQUEMA GENERAL DEL PROYECTO	6
FIGURA 2-1: SISTEMAS DE REFERENCIA GLOBAL (FG) Y LOCAL (FR)	14
FIGURA 2-2: MAPAS GEOMÉTRICOS	16
FIGURA 2-3: MAPAS DE REJILLA	17
FIGURA 2-4: GRAFO Y MAPA TOPOLÓGICO	18
FIGURA 2-5: MAPAS SEMÁNTICOS	18
FIGURA 2-6: GRAFO DE VISIBILIDAD	19
FIGURA 2-7: DIAGRAMA DE VORONOI	22
FIGURA 2-8: CAMINO PROBABILÍSTICO	23
FIGURA 2-9: GRAFO DE CONECTIVIDAD	24
FIGURA 2-10: DESCOMPOSICIÓN EXACTA DE CELDAS	25
FIGURA 2-11: DESCOMPOSICIÓN APROXIMADA DE CELDAS	26
FIGURA 2-12: DESCOMPOSICIÓN ADAPTATIVA	26
FIGURA 2-13: CAMPOS DE POTENCIAL	28
FIGURA 2-14: MÍNIMOS GLOBAL Y LOCAL [NIL09]	29
FIGURA 3-1: VEHÍCULO EZGO	32
FIGURA 3-2: ESQUEMA HARDWARE DEL EQUIPO	34
FIGURA 3-3: LÁSER SICK LMS 291-S05	34
FIGURA 3-4: ESTRUCTURA DE SOPORTE Y LÁSER MONTADO EN EL VEHÍCULO	35
FIGURA 3-5: TARJETAS DE ADQUISICIÓN	36
FIGURA 3-6: PANTALLA TÁCTIL	37
FIGURA 3-7: SISTEMAS IMPLEMENTADOS O EN DESARROLLO	40
FIGURA 3-8: SISTEMA DE SEGURIDAD A DISTANCIA	41
FIGURA 3-9: INTERFAZ TÁCTIL	42
FIGURA 3-10: SISTEMA LÁSER	43
FIGURA 3-11: VISIÓN ESTEREOSCÓPICA	43
FIGURA 3-12: MANDO INALÁMBRICO (Wii)	44
FIGURA 3-13: DISPOSITIVO GPS	45
FIGURA 4-1: PLANO AÉREO DE LA UNIVERSIDAD CARLOS III	49
FIGURA 4-2: LÁSER SICK LMS 291-S05	52
FIGURA 4-3: RAYOS EMITIDO Y REFLEJADO	52
FIGURA 4-4: FUNCIONAMIENTO DEL ESPEJO INTERNO DEL LÁSER [LIDAR]	53
FIGURA 4-5: ÁNGULO DE BARRIDO DEL LÁSER	55

FIGURA 5-1: XSSENS MTi-G	64
FIGURA 5-2: POSICIÓN DE LOS SATÉLITES PARA RECEPCIÓN DE SEÑAL [AAP]	67
FIGURA 5-3: PROBLEMA DEBIDO A LA PRESENCIA DE <i>URBAN CANYONS</i> [AAP]	69
FIGURA 6-1: CLASES QUE COMPONEN EL SISTEMA	78
FIGURA 6-2: ESQUEMA GENERAL DE FUNCIONAMIENTO DE LA CLASE PATHPLANNING	80
FIGURA 7-1: CASOS DE USO DEL VEHÍCULO AUTÓNOMO	87
FIGURA 7-2: SECUENCIA BÁSICA DE PLANIFICACIÓN	88
FIGURA 7-3: IMAGEN COMPLETA DEL CAMPUS	89
FIGURA 7-4: IMAGEN DE LA ZONA DE PRUEBAS	90
FIGURA 7-5: CAMINO GLOBAL PARA DISTINTOS VALORES DE R_G	91
FIGURA 7-6: CAMINOS GLOBALES PARA DISTINTA SEPARACIÓN ENTRE POSICIONES	94
FIGURA 7-7: SEPARACIÓN DE POSICIONES VS NÚMERO DE POSICIONES	94
FIGURA 7-8: SECUENCIA BÁSICA DEL MAPEADO LOCAL	95
FIGURA 7-9: MAPAS LOCALES CON EL LÁSER (A) APUNTANDO AL FRENTE, (B) GIRADO A LA DERECHA, (C) GIRADO A LA IZQUIERDA	96
FIGURA 7-10: MAPAS DE PROXIMIDAD CON EL LÁSER (A) APUNTANDO AL FRENTE, (B) GIRADO A LA DERECHA, (C) GIRADO A LA IZQUIERDA	97
FIGURA 7-11: MAPAS DE MOVIMIENTO CON EL LÁSER (A) APUNTANDO AL FRENTE, (B) GIRADO A LA DERECHA, (C) GIRADO A LA IZQUIERDA	97
FIGURA 7-12: MAPAS INCREMENTALES	98
FIGURA 7-13: MAPAS DE OBSTÁCULOS	98
FIGURA 7-14: MAPAS INCREMENTALES CON EL LÁSER (A) APUNTANDO AL FRENTE, (B) GIRADO A LA DERECHA, (C) GIRADO A LA IZQUIERDA	101
FIGURA 7-15: DERIVA DE LA POSICIÓN, CON EL ROBOT PARADO (A) Y MOVIÉNDOSE EN LÍNEA RECTA (B)	102
FIGURA 7-16: MAPAS DE OBSTÁCULOS CON EL LÁSER (A) APUNTANDO AL FRENTE, (B) GIRADO A LA DERECHA, (C) GIRADO A LA IZQUIERDA	103
FIGURA 7-17: MAPAS DE MOVIMIENTO EN FUNCIÓN DE LA ORIENTACIÓN DEL ROBOT	105
FIGURA 7-18: ESQUEMA DE ALCANCE DE SUBMETA	108
FIGURA 7-19: ESQUEMA DEL TRATAMIENTO DE LAS POSICIONES A ALCANZAR	112
FIGURA 7-20: MAPAS DE OBSTÁCULOS EN DIFERENTES MOMENTOS DEL AVANCE DEL ROBOT	116
FIGURA 7-21: ESQUEMA DE ELIMINACIÓN DE OBSTÁCULOS	117
FIGURA 7-22: ESQUEMA DE ACTUALIZACIÓN DE OBSTÁCULOS	118
FIGURA 8-1: BÚSQUEDA LOCAL – CASO ESTÁTICO (I)	125
FIGURA 8-2: MAPAS DE OBSTÁCULOS – CASO ESTÁTICO (I)	126

FIGURA 8-3: BÚSQUEDA LOCAL – CASO ESTÁTICO (II).....	127
FIGURA 8-4: MAPAS DE OBSTÁCULOS – CASO ESTÁTICO (II).....	129
FIGURA 8-5: MAPA INCREMENTAL – CASO ESTÁTICO (II).....	129
FIGURA 8-6: BÚSQUEDA LOCAL – CASO ESTÁTICO (III).....	130
FIGURA 8-7: EJEMPLOS EXTRAÍDOS DE LA BÚSQUEDA LOCAL – CASO ESTÁTICO (III)	131
FIGURA 8-8: MAPAS DE OBSTÁCULOS – CASO ESTÁTICO (III)	132
FIGURA 8-9: BÚSQUEDA LOCAL – CASO ESTÁTICO (IV)	133
FIGURA 8-10: EJEMPLOS EXTRAÍDOS DE LA BÚSQUEDA LOCAL – CASO ESTÁTICO (IV).....	134
FIGURA 8-11: MAPAS DE OBSTÁCULOS – CASO ESTÁTICO (IV)	135
FIGURA 8-12: MAPA INCREMENTAL – CASO ESTÁTICO (IV)	135
FIGURA 8-13: BÚSQUEDA LOCAL – CASO DINÁMICO (I).....	137
FIGURA 8-14: MAPA DE OBSTÁCULOS AL FINALIZAR LA SIMULACIÓN.....	138
FIGURA 8-15: MAPA INCREMENTAL AL FINALIZAR LA SIMULACIÓN.....	139
FIGURA 8-16: BÚSQUEDA LOCAL – CASO DINÁMICO (II).....	140
FIGURA 8-17: MAPA DE OBSTÁCULOS AL FINALIZAR LA SIMULACIÓN.....	141
FIGURA 8-18: MAPA INCREMENTAL AL FINALIZAR LA SIMULACIÓN.....	141
FIGURA 8-19: BÚSQUEDA LOCAL – CASO DINÁMICO (III).....	143
FIGURA 8-20: MAPA DE OBSTÁCULOS AL FINALIZAR LA SIMULACIÓN.....	144
FIGURA 8-21: MAPA INCREMENTAL AL FINALIZAR LA SIMULACIÓN.....	144
FIGURA 8-22: BÚSQUEDA LOCAL – CASO DINÁMICO (IV)	146
FIGURA 8-23: MAPA DE OBSTÁCULOS AL FINALIZAR LA SIMULACIÓN.....	147
FIGURA 8-24: MAPA INCREMENTAL AL FINALIZAR LA SIMULACIÓN.....	147
FIGURA 8-25: BÚSQUEDA LOCAL – CASO DINÁMICO (V)	149
FIGURA 8-26: MAPA DE OBSTÁCULOS AL FINALIZAR LA SIMULACIÓN.....	150
FIGURA 8-27: MAPA INCREMENTAL AL FINALIZAR LA SIMULACIÓN.....	150
FIGURA 8-28: COMPARATIVA DE PRUEBAS.....	151
FIGURA 8-29: RESULTADOS DE LAS PRUEBAS EN ESTÁTICO	152
FIGURA 8-30: RESULTADOS DE LAS PRUEBAS EN MOVIMIENTO	153
FIGURA 11-1: EJES GEOGRÁFICOS SOBRE MAPA DEL CAMPUS	166
FIGURA 11-2: EJES CARTESIANOS SOBRE MAPA DEL CAMPUS	167
FIGURA 11-3: EJES CARTESIANOS CON ORIGEN LA POSICIÓN DEL ROBOT.....	168
FIGURA 11-4: REFERENCIA GLOBAL Y REFERENCIAS LOCALES	169

Índice de tablas

TABLA 4-1: CARACTERÍSTICAS DEL LÁSER SICK LMS 291-S05	54
TABLA 5-1: SISTEMAS EMPLEADOS EN EL CÁLCULO DE LA INFORMACIÓN (SEGÚN EL MODO DE FUNCIONAMIENTO).....	65
TABLA 7-1: LONGITUD DEL CAMINO ENCONTRADO Y DURACIÓN DE LA BÚSQUEDA PARA DISTINTAS SEPARACIONES ENTRE POSICIONES CONSECUTIVAS.....	93

CAPÍTULO 1. INTRODUCCIÓN

*¿Y no sabes que el principio es lo más
importante en toda obra?*

Platón, *La República*

1. Introducción

El objetivo del presente proyecto consiste en la implementación de un mecanismo de planificación de caminos, formando parte del sistema necesario para dotar de autonomía al robot móvil iCab (Intelligent Campus AutoMoBile). Este prototipo está siendo desarrollado por el Laboratorio de Sistemas Inteligentes (LSI) de la Universidad Carlos III de Madrid. iCab es un vehículo de exteriores, diseñado para el transporte de personas y mercancías dentro del campus de la Universidad Carlos III (campus de la Escuela Politécnica de Leganés).

La realización del proyecto ha necesitado una serie de desarrollos previos al proceso de planificación propiamente dicho, entre las que destacan las relacionadas con la localización del vehículo y el mapeado del entorno; así como ha exigido el empleo de ciertos dispositivos, y la integración de las distintas partes que serán explicadas en el presente documento.

1.1 Necesidades a cubrir en el proyecto

La construcción y puesta en marcha de cualquier dispositivo móvil conlleva una serie de dificultades de distinta índole que son necesarias resolver para el correcto funcionamiento de la aplicación. En el caso a tratar, además de los problemas típicos de movilidad, se plantea la necesidad de analizar e implementar soluciones que permitan la completa autonomía del dispositivo, ya sea en entornos estáticos como en aquellas situaciones en las que elementos externos influyen la reacción del móvil de forma dinámica [Lop08-mov].

El primer punto de estudio consistiría, por tanto, en el análisis de la *geometría* del entorno, es decir, las características del medio donde se desarrollará la actividad del robot. Este esquema incluye las zonas de libre tránsito y los obstáculos fijos, que generan así mismo zonas de visión completa y limitada donde pueden actuar los sensores. Por otra parte, para completar el cuadro general, es necesario un análisis posterior de los elementos dinámicos, esto es, de la naturaleza, tipo y envergadura de

los obstáculos móviles, así como del movimiento de los mismos para estimar la reacción del robot y evitar la colisión.

El estudio de la *cinemática* del robot parte de las posibles operaciones que pueda realizar, y estará limitado por la estructura física del dispositivo. El objetivo de esta parte se centra en resolver el problema del cálculo de rutas, desde un punto inicial hasta alcanzar la meta propuesta, evitando en la medida de lo posible situaciones inesperadas y tratando de optimizar el camino que debe seguir.

Derivado del punto anterior, es necesario incluir un análisis de la *dinámica* del problema, esto es, concretar la reacción del robot ante elementos de aparición puntual que dificulten o impidan el seguimiento de la ruta prefijada. Evitar situaciones conflictivas ante la detección de obstáculos inesperados será uno de los aspectos más importantes y difíciles a plantear, debido a que no se trata de un problema generalizable, y serán necesarias soluciones distintas dependiendo de la situación en cuestión.

Siguiendo estos principios, se puede esbozar una primera aproximación a los escenarios que será preciso tener en cuenta. El primero de ellos y más básico consistirá en un entorno estático con obstáculos fijos, dentro del cual el robot debe ser capaz de calcular la ruta óptima entre dos puntos. Para ello, se precisa una selección adecuada del algoritmo de planificación de caminos, que ha de ser de carácter global y partir del conocimiento completo del entorno.

A continuación, ha de incluirse la presencia de obstáculos no esperados, lo que añade al sistema la necesidad de que el robot reaccione de forma dinámica ante los imprevistos. Dichos cuerpos, en movimiento o estáticos, serán de distinta índole; lo que lleva a considerar no sólo la naturaleza de los mismos, sino también su aparición espontánea en el campo de visión del robot, así como la posibilidad de su conversión posterior a obstáculos fijos.

1.2 Objetivos a desarrollar

Las características del escenario donde se desarrolla la actividad del robot impiden la aplicación únicamente de un algoritmo de planificación global, debido a la presencia de elementos dinámicos que dificultarían e incluso impedirían el seguimiento de la ruta estática por parte del móvil.

Parecería más conveniente prescindir de la información previa del entorno y centrar la implementación en el desarrollo de un algoritmo de planificación local que calcule la ruta a seguir según los parámetros proporcionados por los sensores. Estos métodos, aunque de demostrada eficacia en ciertas situaciones, adolecen de los problemas típicos de la falta completa de información, pudiendo no resultar óptimos e incluso, no converger a la solución requerida.

La propuesta a desarrollar debe aunar en la medida de lo posible las ventajas proporcionadas por ambas familias de métodos, e intentar minimizar los inconvenientes particulares de cada una de ellas mediante un respaldo mutuo.

En primer lugar, se sugiere la existencia de un entorno conocido en cuanto a la situación de obstáculos permanentes. En el caso que nos ocupa, este escenario será el propio patio de la Universidad. Es por tanto necesario, en posteriores implementaciones, un análisis del medio particular donde circulará el móvil. Dicho entorno se dividirá en partes elementales ó células para facilitar los cálculos.

Inicialmente, sobre dicho escenario se construirá un mapa del entorno siguiendo las premisas de campo de potencial, esto es, dotando a las zonas libres (donde es posible la circulación del robot) de una fuerza atractiva, que será mayor cuanto más próxima se encuentre dicha zona respecto al punto destino; dicho objetivo final dispondrá de un nivel mayor de fuerza atractiva que atraerá el móvil hacia sí. En cuanto al obstáculo fijo (zona donde no está permitido el movimiento del móvil), se modelará como fuente de fuerza repulsiva, rechazando el movimiento del robot hacia su entorno y forzando al mismo a su encaminamiento hacia zonas de potencial más positivo. El disponer de una estructura de celdas permitirá delimitar con más exactitud las distintas regiones existentes [Lop08-mov].

Sobre este plano estático se calcularán las posibles rutas permitidas y se elegirá la óptima según distancia recorrida. Para ello, será necesario aplicar un algoritmo de planificación global, otorgando a cada celda de potencial atractivo un coste asociado; el camino óptimo estará constituido por aquellas posiciones libres consecutivas que minimicen la función de costes.

Con la configuración explicada, es ya posible permitir que el robot vaya recorriendo el espacio de trabajo. A medida que recorra la ruta marcada, irán apareciendo nuevos obstáculos que serán detectados por medio de sensores. En dichas situaciones, es posible que el móvil deba recalcular la ruta más próxima a la original, modificando su trayectoria e intentando volver al camino inicial en cuanto se den las condiciones adecuadas (cuando haya rodeado al obstáculo o este ya no se encuentre obstruyendo el camino inicial).

En esta parte radica una de las mayores dificultades del estudio, debido a la limitación temporal a la que está sometida la reacción del móvil y el número finito de opciones que es posible barajar. Ello llevará a analizar la aportación de los distintos parámetros en el resultado final (velocidad del móvil, ángulo del eje principal, dirección y sentido de la desviación...) para valorar la influencia de los mismos y limitar el muestreo en el espacio de posibles soluciones.

1.3 Fases del proyecto

La Figura 1-1 resume de forma general las pautas desarrolladas a lo largo del presente proyecto.

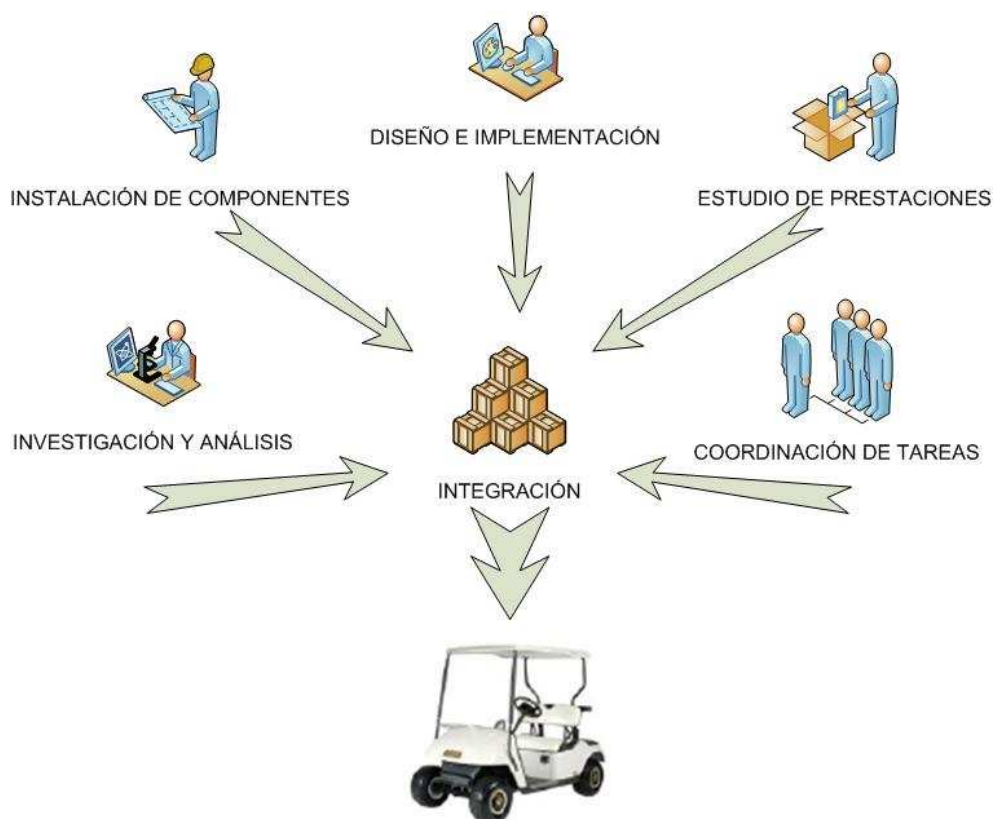


Figura 1-1: esquema general del proyecto

El paso inicial consistió en la investigación y análisis del problema, teniendo en cuenta situaciones anteriores que podían ser de utilidad o se asemejaban a la actual, así como las posibles soluciones. Fue necesario un estudio previo de los medios disponibles en el laboratorio y los desarrollos efectuados hasta el momento, así como el análisis de los distintos métodos de planificación y las opciones existentes y utilizables. Así mismo, se hizo necesario un proceso de familiarización con la plataforma, tanto hardware como software.

Una vez conocidos el alcance de los medios disponibles, y habiendo adquirido la capacidad de emplearlos de forma adecuada, se hizo necesaria la instalación y puesta a punto de los componentes, tanto los dispositivos físicos como sus interfaces y drivers, bibliotecas de desarrollo y programas necesarios.

Disponiendo de las herramientas y de un conocimiento adecuado de las mismas, era preciso saber qué hacer con ellas y cómo enfocar la situación planteada;

esto derivó en el diseño de una estrategia de actuación, esto es, la secuencia de pasos a desarrollar para alcanzar el objetivo marcado. Una vez conocidas las acciones necesarias, era precisa implementarlas.

Había que comprobar que los pasos desarrollados eran los correctos, y los resultados obtenidos los esperados; por lo que fue necesario tratar de forma individual los distintos bloques que componen la aplicación, forzando las condiciones de entrada - en caso necesario- para comprobar las salidas.

El siguiente paso a desarrollar consistió en coordinar las distintas tareas, de forma que aquellas que fuesen dependientes interactuasen entre sí de forma correcta.

Finalmente, se realizó la integración de las distintas partes implicadas y su posterior ajuste, validándose la implementación mediante una tanda de pruebas para comprobar las prestaciones del sistema completo.

1.4 Problemas en la planificación de caminos

El proceso de planificación se inició mediante la modelización del entorno global del medio para obtener la trayectoria completa, uniendo la posición origen con la de destino, y dando como resultado una serie de submetas a alcanzar necesarias para la posterior búsqueda local.

Una vez obtenidos estos datos, fue necesario construir un modelo del entorno próximo al móvil mediante la información proporcionada por los sensores, teniendo en cuenta tanto la información sobre los obstáculos como la referente a la posición del dispositivo en el medio. Por tanto, no se trató de un proceso de construcción directa, sino que necesitó ciertos pasos intermedios para su consecución, los cuales serán explicados y analizados a lo largo del presente documento.

Desde la perspectiva del mapeado, existen diversas razones de carácter general que dificultan la adquisición de un modelo para vehículos móviles [Thr05], entre las que cabe destacar:

- La enorme dimensión del espacio de hipótesis, esto es, de todos los posibles mapas existentes. Ya que los mapas son definidos en un espacio continuo, y aún en el caso de realizar aproximaciones discretas, la cantidad de variables implícitas en los cálculos resulta demasiado elevada, hasta el punto de resultar inviables su cálculo.
- Problemas derivados de la localización espacial del robot. En primer lugar, determinar su situación en entornos móviles deriva normalmente en errores de odometría que se van acumulando hasta, llegado el momento, dar lugar a soluciones completamente erróneas. En segundo lugar, el problema del mapeado conlleva la necesidad no sólo del tratamiento correcto de la información del entorno, sino también de la elección adecuada de la representación y tratamiento del mismo. La dependencia mutua entre ambos problemas a tratar dificulta su tratamiento e implementación.

Además, existen una serie de dificultades dependiendo de la implementación desarrollada, que dependen de factores variables, y cuya solución dependerá sobre todo de las características propias del problema [Thr05]:

- El tamaño del entorno donde se desarrolla la actividad del móvil, relacionado de forma directa con el alcance máximo del sensor de medida y su fiabilidad. En el caso de la construcción de mapas incrementales, este problema se agrava a medida que el robot se desplaza, al aumentar el coste computacional y los errores odométricos.
- Las condiciones del entorno y el ruido en percepción y actuación, capaces de derivar en datos erróneos que desvíen la solución calculada de la real. No sólo existe la dificultad de detectar posibles errores, sino también la de rectificarlos de forma conveniente.
- La ambigüedad en los datos percibidos, debido a los cambios en el escenario, y su correcto tratamiento.
- Los ciclos en el entorno, particularmente difíciles de mapear, cuando el móvil recorre el mismo camino en distinto sentido, y que puede derivar en rutas distintas bajo las mismas condiciones.

- La pérdida de información al modelar espacios tridimensionales como planos bidimensionales. Los sistemas de percepción no serán capaces de evaluar todas y cada una de las características del medio.

En la solución presentada a continuación, se tratarán de resolver las dificultades comentadas en la medida de lo posible. Se propondrá un sistema adecuado de obtención de datos, desarrollándose posteriormente mecanismos que permitan reducir el volumen de esta información.

Por otro lado, se evitarán los problemas derivados de la posición del móvil mediante el empleo de dispositivos adecuados y la separación de las funciones de percepción y localización. De esta forma, el problema se reduce a lo que es llamado *mapping with known poses* [Thr05].

Con la información obtenida, se tratará la problemática del mapeado y se estudiará el empleo adecuado de algoritmos de búsqueda de caminos.

1.5 Estructura del documento

El presente documento se desarrolla en nueve apartados, el primero de los cuales está formado por la presente introducción, donde se introducen las necesidades, objetivos y fases del proyecto, así como una visión general de la problemática asociada a la planificación.

El apartado 2 desarrolla la teoría previa a la implementación del sistema, donde se estudian distintos métodos de mapeado y planificación, para elegir el que se adecúe a las necesidades del proyecto.

El apartado 3 describe el entorno de desarrollo y la plataforma empleada, así como los componentes ya presentes en el modelo iCab.

El apartado 4 describe el sistema de percepción del entorno y las características con las que se modela el mismo, base fundamental para el proceso de mapeado.

El apartado 5 trata sobre el sistema de localización del vehículo en su entorno.

El apartado 6 desarrolla la programación presente en el proyecto, tanto el uso de distintas librerías como la implementación de clases propias.

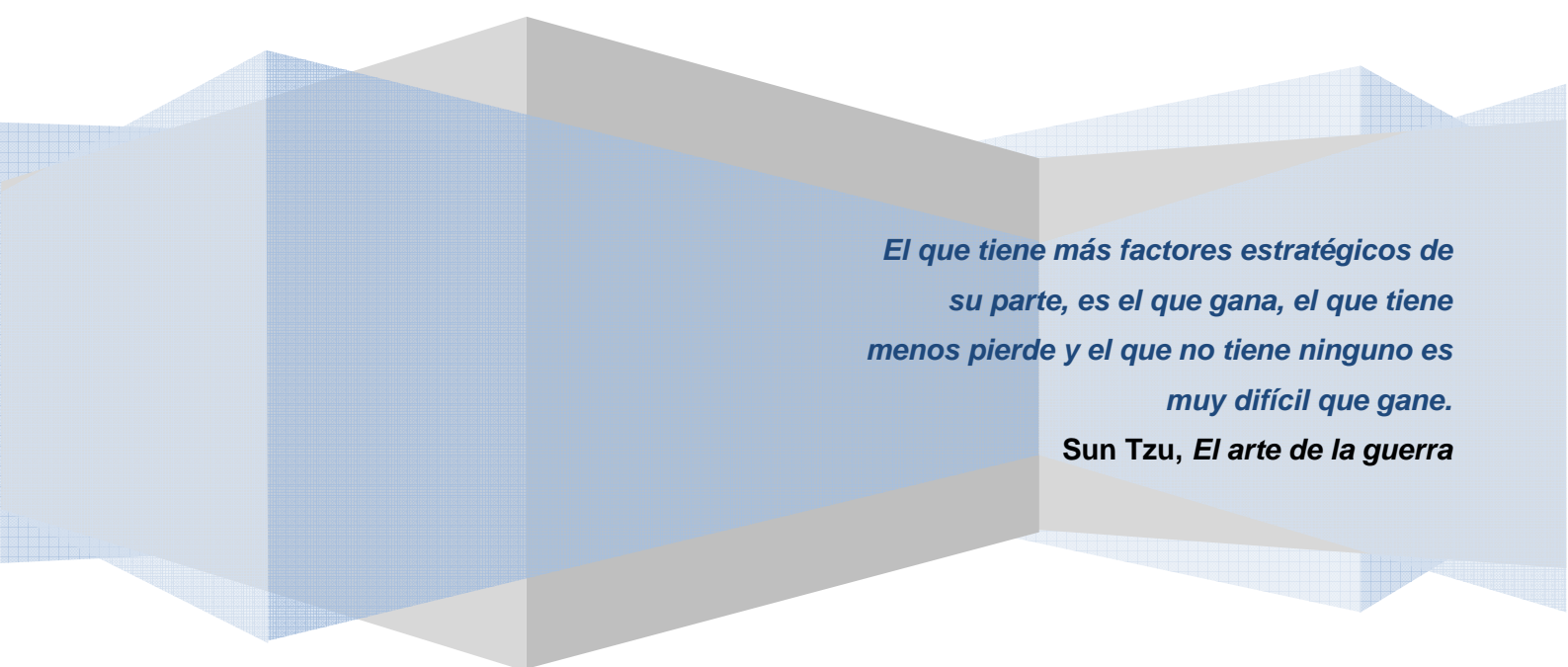
En el apartado 7 se documenta la implementación realizada. Dicho apartado comienza introduciendo la base teórica presente en los mapas de ocupación probabilística, empleados en el proyecto, para continuar con el proceso de integración de los distintos componentes presentes. Posteriormente, se analiza el problema de la planificación de dos formas distintas: global y local; para finalizar con el análisis del mapeado incremental del entorno a medida que el vehículo se desplaza por el medio.

El apartado 8 muestra los resultados obtenidos en distintos casos y bajo circunstancias diferentes, tratando de identificar características y patrones significativos, así como probando el correcto funcionamiento del sistema.

El apartado 9 desarrolla las conclusiones obtenidas del proyecto, así como las posibles líneas de investigación derivadas del mismo y mejoras futuras.

CAPÍTULO 2.

ESTADO DEL ARTE



El que tiene más factores estratégicos de su parte, es el que gana, el que tiene menos pierde y el que no tiene ninguno es muy difícil que gane.

Sun Tzu, *El arte de la guerra*

2. Estado del arte

2.1 Introducción al concepto de robots móviles autónomos

El concepto de *movilidad* surge de la capacidad del robot para realizar movimientos, ya sea mediante su traslación o cambio de posición, o mediante la rotación del dispositivo en sí o de alguna de sus partes. La primera de estas situaciones implica un desplazamiento, y es la que se asocia comúnmente al concepto de robot móvil; este comportamiento no tiene que estar ligado obligatoriamente a un proceso de razonamiento, pudiendo realizarse de forma predefinida. En el caso de que el movimiento del robot implique una lógica adicional traducida en operaciones de acción, percepción y reacción, es cuando hablamos de *autonomía*: capacidad que permite al dispositivo tomar decisiones en función de las circunstancias en las que se encuentre y de la información previa suministrada por los sensores, definiendo así las estrategias de actuación [Cas02] [Muñ95].

Por tanto, al hablar de un robot autónomo, se hace necesario el análisis del entorno y de las características cinemáticas y dinámicas del problema, así como un proceso de planificación que evalúe las posiciones a alcanzar y los movimientos asociados. Este es uno de los aspectos más importantes a cubrir en robótica móvil autónoma, y es lo que se conoce como *navegación*.

Navegación

La *navegación* es la técnica que permite a un móvil desplazarse de un punto origen a otro destino, a través de un entorno con obstáculos, donde el medio puede ser no estructurado y/o desconocido. Esto deriva en una serie de procesos vinculados a la consecución del camino:

- Proceso de percepción: reconocimiento de elementos característicos del entorno que permiten obtener un modelo del mismo, mediante el empleo de información previa y/o de la proporcionada por sensores externos.
- Proceso de planificación: a partir del modelo anterior y empleando una estrategia adecuada, obtención de un camino libre de obstáculos que lleve

a la meta (dicho recorrido puede estar a su vez formado por sub-objetivos intermedios), así como de las trayectorias a seguir.

- Proceso de actuación ó reacción: seguimiento del camino y reajuste de la ruta seguida en función de los cambios en el entorno, lo que implica hacer frente a circunstancias imprevistas en tiempo real.

Planificación

La *planificación de caminos*, que aúna la generación del recorrido y la planificación de trayectorias, es uno de los aspectos críticos del proceso de navegación. Dependiendo del enfoque que adopte el modelo, se pueden distinguir dos tipos de planificadores:

- Planificador global: construye un modelo del entorno con información a priori sobre el mismo, a partir del cual calcula el camino libre de obstáculos mediante la aproximación a una serie de submetas a alcanzar. Su principal problema es la sensibilidad a cambios en el medio y la incapacidad de adaptación a los mismos. Se lleva a cabo antes de cualquier inicio de movimiento, y es útil en entornos controlados y conocidos, como pueden ser interiores.
- Planificador local o reactivo: construye un modelo del entorno con información proporcionada por los sensores in situ, con lo cual proporciona mayor fiabilidad ante el medio donde se encuentra el robot y los cambios que puedan producirse. Su principal problema radica en la falta de generalidad del modelo, pudiendo no abarcar la meta a alcanzar dentro del margen de medida de los sensores. Se lleva a cabo en tiempo de ejecución.

Resulta interesante hacer una distinción entre el concepto de camino y el de trayectoria [Pue07]. El camino, como se ha indicado, hace referencia al recorrido o ruta a seguir por el móvil; mientras que la trayectoria implica el desarrollo de un desplazamiento, por lo que resulta más conveniente emplearla para indicar el movimiento del robot entre dos puntos. Sin embargo, ambas terminologías se emplearán indistintamente a lo largo del presente documento

2.2 Métodos clásicos de planificación

Previamente al análisis de los distintos métodos existentes en la planificación de caminos, es necesario definir una serie de conceptos necesarios para entender los objetivos que se persiguen con la aplicación de los mismos.

En primer lugar, partimos del robot móvil como actor principal, y del entorno en el que se encuentra como escenario de las actuaciones del mismo, conocido como espacio de trabajo.

La interacción entre ambos elementos conforma la configuración existente, esto es, la posición y orientación del móvil en el espacio de trabajo [Muñ95]. Desde un punto de vista bidimensional, el entorno se puede definir mediante un sistema de referencia global F_g , mientras que el robot se encuentra asociado a su propio sistema de coordenadas local F_r (ver Figura 2-1); la relación entre ambos sistemas determina la configuración q del robot en el medio, y puede formalizarse mediante su posición y orientación, de la siguiente manera:

$$q = (p, \theta) = (x, y, \theta)$$

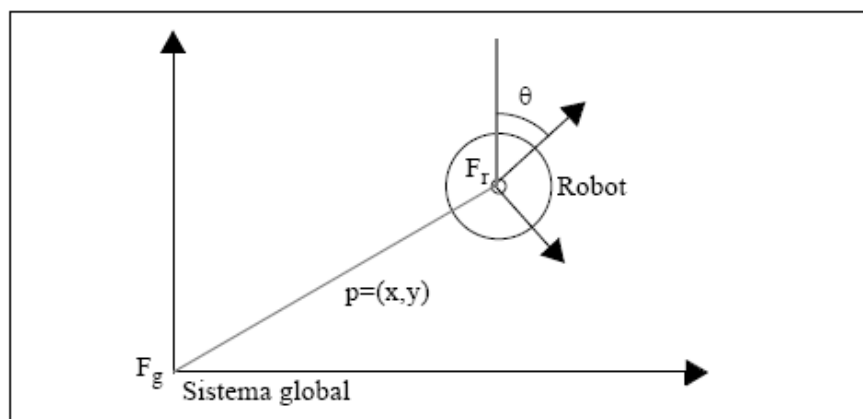


Figura 2-1: sistemas de referencia global (F_g) y local (F_r)

Así, se podría definir el espacio de trabajo como el conjunto de configuraciones que el móvil puede tomar. De forma elemental, dicha configuración dispondrá de dos estados en cuanto a capacidad de movimiento se refiere: cuando el

robot no colisiona con ningún obstáculo (se encuentra situado en el espacio libre C) se halla entonces en configuración libre; en caso contrario, la configuración está en colisión. La planificación consiste pues en encontrar una serie de configuraciones q_i sucesivas pertenecientes al espacio libre C , tales que permitan conectar una configuración inicial q_a con otra final q_f .

2.3 Construcción de mapas

El mapa representa un modelo del espacio de trabajo, donde quedan perfectamente definidas las regiones libres y las ocupadas por obstáculos.

Una primera clasificación de los tipos de mapas existentes se centra en el ámbito que abarcan. Así, pueden ser clasificados como globales (cuando representan el entorno completo, normalmente mediante el empleo de la información existente sobre el mismo) o como locales (cuando muestran el entorno próximo al móvil, construido a partir de las medidas proporcionadas por sensores en tiempo de ejecución)

Una división posterior de los tipos de mapas se orienta en los niveles de representación de la información: *métrico*, *topológico* y *semántico*. A continuación se exponen dichos tipos y sus características principales [Muñ95], [San04], [Cañ07], [Pue07], [Lop08-mov], [Gal07], [Suc09].

2.3.1 Mapas métricos

Representan la ocupación del entorno mediante medidas en el espacio de trabajo, con la consecuente modelización de los objetos (obstáculos) presentes en él. Los mapas obtenidos a nivel métrico se pueden clasificar a su vez en dos grandes familias: los de representación geométrica y los de descomposición espacial (mapas de rejilla).

Mapas geométricos

Los mapas geométricos representan los obstáculos mediante figuras geométricas en 2 o 3 dimensiones (Figura 2-2). En el caso del mapa plano, por ejemplo, se emplean puntos, rectas, círculos...construidos a partir de los datos de medida.

Presentan la ventaja de proporcionar un modelo básico del entorno, facilitando la detección de obstáculos con un bajo coste computacional. Su empleo es conveniente en entornos donde existan obstáculos grandes y con una definición clara, y no resultan óptimos cuando la densidad de objetos pequeños es mayor. Otra dificultad reside en la incorporación de nuevos datos, lo que plantea serios problemas en entornos dinámicos

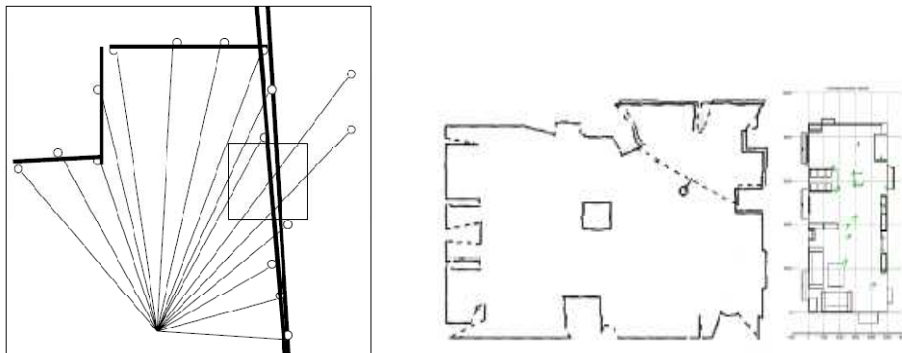


Figura 2-2: mapas geométricos

Mapas de rejilla

Los mapas de rejillas proceden de la descomposición del espacio en unidades mínimas o celdas mediante el empleo de rejillas de ocupación (Figura 2-3). Existen diversas clases de celdas en función de su naturaleza, lo que permite construir diferentes tipos de mapas de rejilla. Así, en función del tratamiento que reciben, las celdas pueden ser binarias o probabilísticas; en función de su tamaño y de la relación entre ellas, se clasifican en uniformes o jerárquicas.

Este tipo de mapas presenta ciertas ventajas en cuanto a su construcción, ya que la estructura discreta del conjunto de celdas permite introducir con facilidad información referente a la posición del móvil y a las mediciones tomadas, incluso cuando se producen desplazamientos y cambios de orientación. Sin embargo, la misma segmentación del espacio provoca un aumento de la carga computacional y de la complejidad en el tratamiento de los datos.

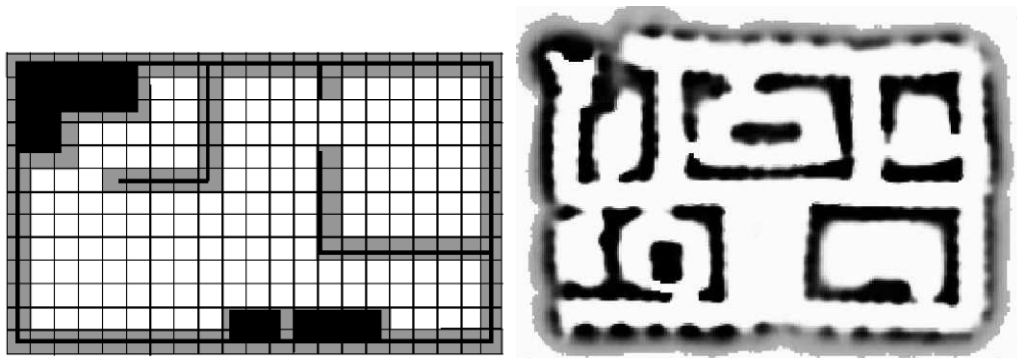


Figura 2-3: mapas de rejilla

2.3.2 Mapas topológicos

Se basan en las relaciones existentes entre las características del entorno, más que en su posición absoluta. En este tipo de mapas, se definen una serie de posiciones en el espacio libre, denominadas nodos, y se representan las conexiones entre ellos mediante el empleo de arcos (Figura 2-4). El conjunto de posiciones y de las relaciones entre ellas permite la construcción de un grafo, al que es posible añadir información métrica (longitud y orientación de los arcos, coste de alcanzar la siguiente posición)

Son mapas que ofrecen beneficios en cuanto al cálculo de trayectorias, acotándola a la búsqueda de caminos en el grafo, y no requieren un conocimiento exacto de la posición del móvil; aunque sí suelen necesitar una información más amplia que la proporcionada por los sensores, por lo que se suelen emplear en entornos conocidos y modelos de planificación global.

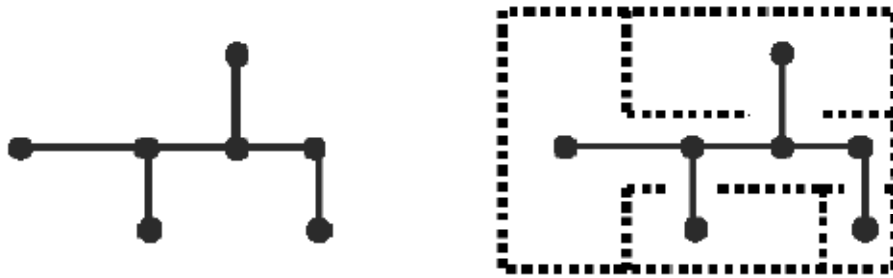


Figura 2-4: grafo y mapa topológico

2.3.3 Mapas semánticos

Se trata de una representación abstracta de las propiedades del entorno. Las partes más representativas del mismo reciben una etiqueta o nombre que los identifica (*pared*, *puerta*, *pasillo*), de forma que es posible la clasificación de los elementos según la información previa almacenada sobre los distintos tipos de elementos (Figura 2-5). De esta forma, se obtiene un resultado mucho más similar a la comunicación humana que en el caso de los modelos anteriormente expuestos.

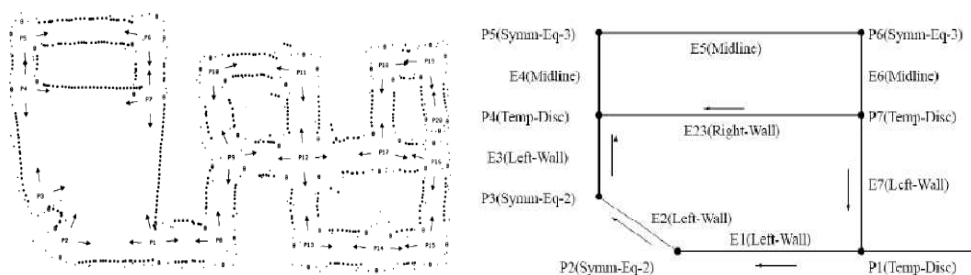


Figura 2-5: mapas semánticos

2.4 Mapas de carretera (Roadmaps)

2.4.1 Grafo de visibilidad

Este tipo de planificador se basa en el llamado concepto de *visibilidad*. Partiendo de un espacio bidimensional y modelando los obstáculos como polígonos, se definen los puntos visibles como aquellos que se pueden unir mediante un segmento continuo perteneciente al espacio libre del entorno C , es decir, sin interceptar ninguno de dichos obstáculos. Para ello, se consideran puntos visibles las aristas de los polígonos que representan las zonas de colisión. El grafo se construye eligiendo los segmentos rectilíneos que, uniendo los puntos visibles del entorno desde la posición inicial a la final (ambas a su vez puntos visibles), permiten optimizar cierta función de costes.

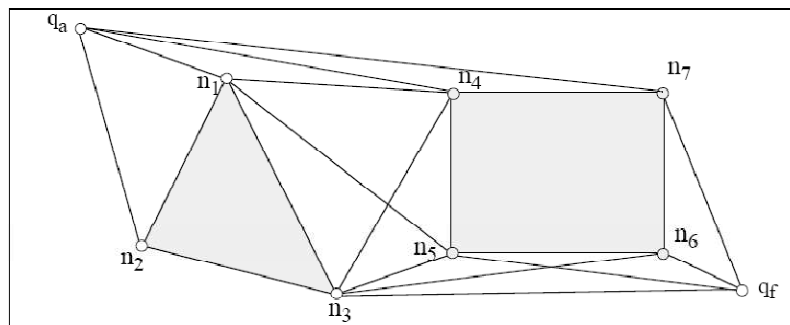


Figura 2-6: grafo de visibilidad

La Figura 2-6 representa un entorno bidimensional con dos obstáculos, donde q_a y q_f son las posiciones inicial y final, respectivamente, y las n_i representan los puntos visibles tratados por el algoritmo de búsqueda. Como se puede comprobar, cada punto se encuentra conectado con aquellos con los que existe visibilidad directa, lo que hace que los segmentos obtenidos se incluyan las aristas de los propios obstáculos.

Entre las ventajas de emplear este tipo de planificadores, se encuentran:

- Admiten el empleo de algoritmos de optimización para mejorar las prestaciones del grafo.
- Coste computacional relativamente bajo.

Sin embargo, existen una serie de inconvenientes que pueden hacer inviable la aplicación de este método en ciertos entornos, como son:

- Es necesario disponer del conocimiento global del entorno, esto es, de la forma, tamaño y situación de los obstáculos.
- En entornos conocidos, existe una mala adaptación a cambios en el mismo.
- Al ser necesario considerar todas las opciones de caminos posibles, el algoritmo puede aumentar su complejidad dependiendo de la cantidad de obstáculos involucrados y de la forma de los mismos.
- Existe un problema adicional al incluir las aristas del obstáculo como segmentos de unión entre dos puntos visibles: es necesario considerar el móvil como elemento puntual (cuando en la práctica no lo será). Los caminos así obtenidos reciben el nombre de rutas semi-libres.(según algunos autores)

Para solventar algunas de las dificultades anteriores, existen una serie de alternativas aplicables al algoritmo base:

- No generar el árbol completo, sino el subgrafo necesario para encontrar el objetivo, obteniendo una ruta que, aun no siendo la óptima, proporcione una mayor rapidez de cálculo y disminuya la complejidad.
- Eliminar la extrema proximidad a las zonas de no transito mediante una ampliación de la superficie ocupada por los obstáculos fijos, lo que se conoce como entorno expandido. Otra opción similar es el modelado circular del elemento móvil.

- En entornos no conocidos, existe la posibilidad de emplear como planificador local una variante denominada LMAV ([Rao98], 1998). Dicho algoritmo procesa los puntos que son visibles desde la posición en la que se encuentra el móvil, seleccionando el más cercano (menor distancia euclídea) respecto a la posición final, y desplazándose hacia él. Se produce la iteración de este proceso hasta llegar al destino (éxito) o hasta que no quedan más nodos visibles por visitar (fracaso).

2.4.2 Diagrama de Voronoi

Al igual que el planificador anterior, se trata de un método basado en el concepto de visibilidad y aplicable principalmente a entornos conocidos bidimensionales.

Sin embargo, el principio de funcionamiento es opuesto al presentado en el punto anterior, ya que en este caso los nodos visibles serán aquellos situados en la posición más alejada del obstáculo (puntos equidistantes entre los dos obstáculos más próximos)

Esto da lugar a la existencia de dos tipos de segmentos: rectilíneos y parabólicos, dependiendo de la estructura del problema. De esta forma, los puntos visibles situados entre dos aristas de obstáculos se unirán mediante una línea recta; mientras que si se encuentran en zonas donde existen vértices suele ser más adecuado el empleo de parábolas.

En la Figura 2-7 se representa un entorno bidimensional delimitado por el polígono de aristas e_i donde existe un único obstáculo, de vértices n_i y aristas a_i . Se puede comprobar la existencia de los dos tipos de segmentos que constituyen el diagrama de Voronoi C_v : los de tipo parabólico como S_1 , situado entre la arista e_1 y el vértice n_2 ; y los de tipo rectilíneos como S_2 , situado entre las arista e_1 y e_2 .

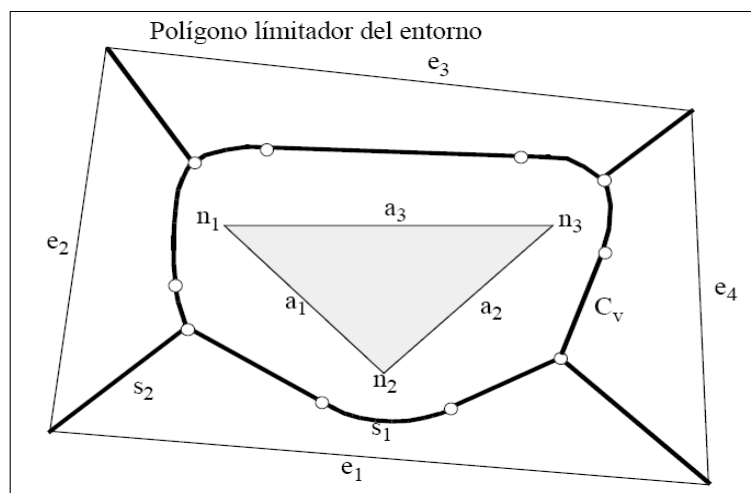


Figura 2-7: diagrama de Voronoi

2.4.3 Caminos probabilísticos (PRM)

El objetivo de este planificador es la generación del mapa (grafo) de forma probabilística en el espacio libre C . Su desarrollo consta de dos fases diferenciadas, que son la de construcción y la de consulta. En ciertas situaciones, existe un último paso antes de dar por concluida el cálculo de la ruta, llamada fase de expansión.

En la fase de construcción se realiza la llamada generación de configuraciones, consistente en generar n posiciones libres de obstáculos de manera aleatoria uniforme en todo el espacio de trabajo. La cantidad de configuraciones es uno de los parámetros a tratar dependiendo del tipo de problema, siendo necesario su ajuste para asegurar la conectividad del grafo generado. Posteriormente, se realiza la conexión de cada nodo con sus k vecinos más cercanos (de nuevo siendo k otro de los parámetros a ajustar), en la medida en que esas conexiones sean posibles, formando una red de posibles rutas a seguir por el móvil o grafo representativo del espacio libre C .

En la fase de consulta, se realiza el cálculo de la trayectoria entre las configuraciones inicial y final, primero conectándolas a alguno de los nodos obtenidos de forma aleatoria, para posteriormente realizar la búsqueda de ruta en el grafo.

La Figura 2-8 muestra un ejemplo de grafo obtenido mediante el empleo del planificador probabilístico. Los nodos son determinados de forma aleatoria dentro de la

zona de libre tránsito, y conectados con sus otros nodos cercanos. Se puede comprobar que el número de conexiones varía dependiendo de la posición del nodo respecto a sus vecinos.

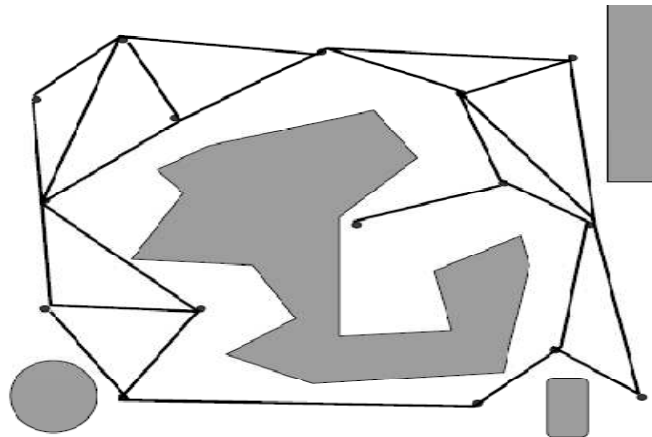


Figura 2-8: camino probabilístico

A pesar de tratarse de un método eficiente y relativamente rápido, adolece de ciertas dificultades relacionadas con su carácter probabilístico, como pueden ser la posibilidad de una alta concentración de nodos en una misma zona o una cantidad insuficiente de ellos en zonas de tránsito estrechas. En este caso se realizaría la fase de expansión, consistente en localizar las zonas de difícil acceso y no conectadas, expandiendo el grafo para unir las con el resto.

Otro aspecto crítico es la forma en la que se realiza la conexión entre los nodos, debido al tiempo que requiere; se suele emplear para ello un planificador local determinista, ya que proporciona una respuesta eficiente. Además resulta muy dependiente de la métrica empleada y es factible la aparición de lazos o caminos de mayor recorrido.

2.5 Descomposición en celdas

Este mecanismo basa su funcionamiento en la división del espacio libre en celdas no solapadas. De esta forma, el planificador busca la ruta que una los puntos

inicial y final (que a su vez serán modelados como celdas) mediante la sucesión de celdas adyacentes entre sí libres de obstáculos.

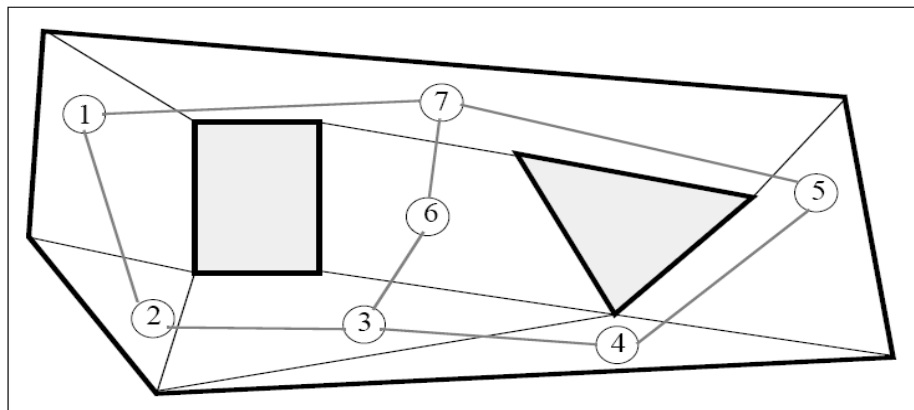


Figura 2-9: grafo de conectividad

En este proceso, al contrario que en los dos anteriores, no existe la creación de segmentos que unan posiciones del espacio, sino que la ruta estará formada por una sucesión de celdas, lo que hace necesario la construcción posterior de un grafo de conectividad (no dirigido) que defina el camino, como se puede observar en la Figura 2-9, donde cada celda representa un nodo y los arcos del grafo unen aquellos nodos que son adyacentes.

Tras la obtención del grafo de conectividad, es necesario aplicar un algoritmo de búsqueda que una las posiciones inicial y final, y que defina la ruta a seguir (las celdas por las que ha de pasar el móvil).

La descomposición en celdas plantea ciertas cuestiones que dependen del entorno concreto, como son la elección de una adecuada forma geométrica para las mismas que facilite el cálculo de distintas configuraciones y asegure la adyacencia de las celdas. Para resolver distintas situaciones se pueden plantear distintas variantes del planificador:

- Descomposición exacta: el espacio libre C está representado por una colección de celdas no traslapadas cuya unión es exactamente C (el espacio libre se halla completamente dividido en celdas).

Se trata de un método completo (es decir, encuentra solución si esta existe) Ejemplos de este método son las descomposiciones trapezoidales y las cilíndricas. En la Figura 2-10 se puede observar un entorno delimitado por un polígono donde existen dos obstáculos; el espacio libre es completamente dividido en celdas trapezoidales y triangulares construidas mediante líneas verticales que parten de cada vértice, y donde las celdas adyacentes comparten una arista común.

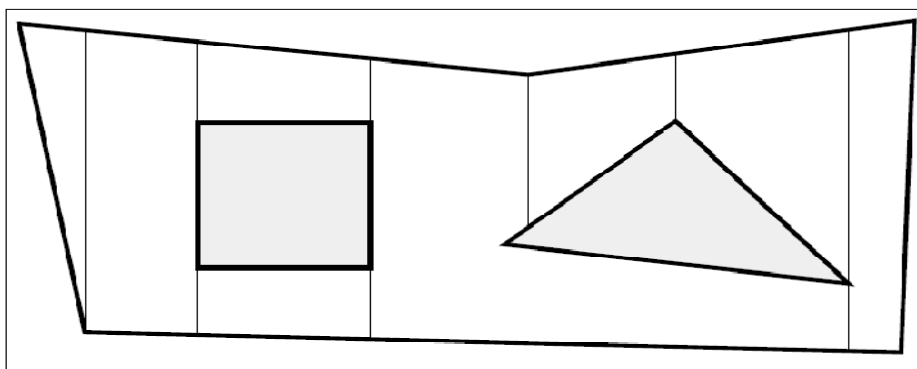


Figura 2-10: descomposición exacta de celdas

- Descomposición aproximada: el espacio libre C está representado por una colección de celdas no traslapadas cuya unión está incluida en C (el conjunto de todas las celdas no corresponde exactamente al total del espacio libre).

Se trata de un método no completo pero con un cálculo más rápido de las trayectorias.

Dentro de esta aproximación, el espacio libre puede ser modelado mediante descomposición fija o mediante descomposición adaptativa.

En la descomposición fija, el espacio se descompone en celdas de igual tamaño, que podrán estar ocupadas por obstáculos o representar al espacio libre. En la Figura 2-11 se pueden observar los principios de esta descomposición, y como el resultado de la misma es una aproximación al entorno real del problema.

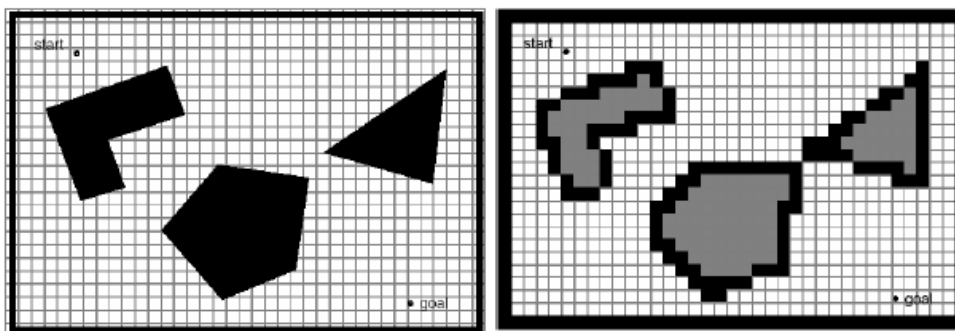


Figura 2-11: descomposición aproximada de celdas

Ejemplos de descomposición adaptativa son los quadrees (división de celdas en cuatro partes) y los octrees (división de celdas en ocho partes). La Figura 2-12 representa este tipo de descomposición del espacio libre, donde el rectángulo que delimita el espacio es recursivamente dividido en rectángulos más pequeños mientras que en las celdas obtenidas sigan existiendo zonas de no tránsito, y hasta llegar a una resolución determinada.

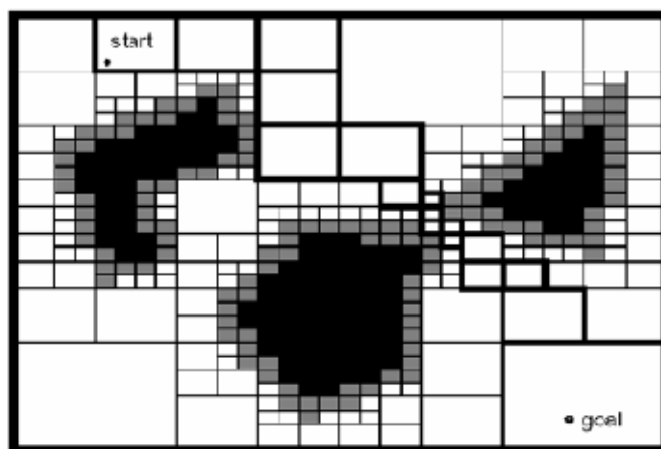


Figura 2-12: descomposición adaptativa

2.6 Campos de potencial

Se trata de un método de planificación local en entornos desconocidos, por lo que difiere de los planteamientos hasta ahora expuestos, al basarse en técnicas reactivas de navegación.

Este planificador considera el móvil bajo la influencia de un campo de potencial artificial, evaluando su posición en todo punto del ambiente conocido mediante la influencia y aproximación de los obstáculos. Consiste en representar el ambiente en un punto concreto p del espacio mediante una función de potencial U , definida sobre el espacio libre, y compuesta por la unión de un potencial atractivo U_a (generado por el destino) y un potencial repulsivo U_r (generado por los obstáculos), es decir:

$$U(p) = U_a(p) + U_r(p)$$

Debido a este potencial, sobre el móvil actúa una fuerza artificial en el punto p , que es la siguiente:

$$F(p) = -\nabla U(p)$$

Al igual que la función potencial, la fuerza sobre el móvil se puede descomponer en una fuerza atractiva que le dirige hacia el destino y una repulsiva que le aleja de los obstáculos:

$$F(p) = F_a(p) + F_r(p)$$

De esta forma, los pasos a seguir en el desarrollo del algoritmo se resumen en:

- Calcular el potencial en un punto $U(p)$ según información de los sensores
- Calcular la fuerza sobre el móvil $F(p)$ en ese punto
- Hacer que el móvil actúe según los parámetros calculados (movimiento en la dirección y sentido marcados por $F(p)$)

Las fuerzas atractivas y repulsivas se pueden definir de distintas formas. La primera y más básica de ellas consiste en medir la fuerza atractiva en función de la distancia euclídea al destino, de manera que cuando el móvil se aproxime a dicho punto, disminuya su influencia; mientras que la fuerza repulsiva sólo influya cuando el móvil está próximo ó se dirija a un obstáculo, alejándolo del mismo. En la posición destino, ambas fuerzas deben anularse.

En la Figura 2-13 se pueden observar la aplicación de esta metodología en el espacio definido en (a). El destino posee carácter atractivo que orienta el movimiento hacia sí, mientras que los obstáculos presentan un potencial repulsivo que evita el

movimiento hacia ellos, como se puede comprobar en (b) y (c), respectivamente. El mapa final es la combinación de ambos potenciales, dando lugar a (d).

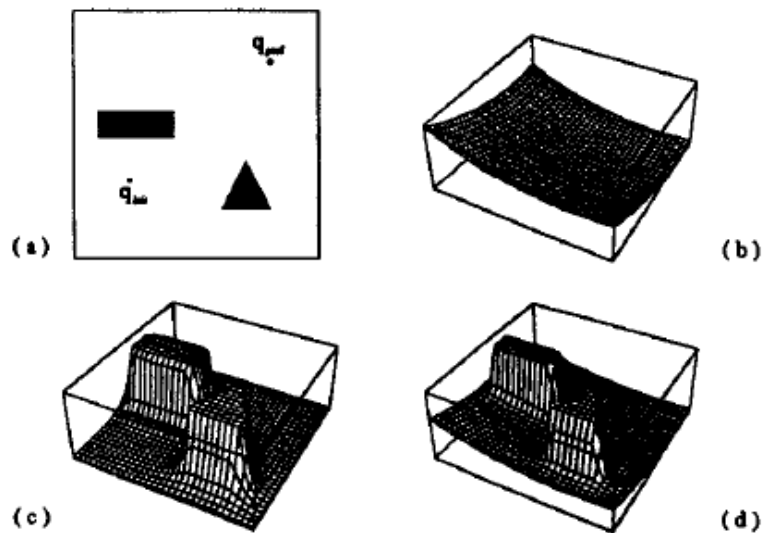


Figura 2-13: campos de potencial

Su gran ventaja radica en la no necesidad de poseer un conocimiento completo del entorno, lo que resulta especialmente útil en entornos dinámicos y cambiantes. Además, proporciona soluciones deterministas, al estar basado en potenciales artificiales (no en magnitudes físicas).

Entre sus inconvenientes se pueden citar:

- Existe una alta dependencia de la definición/magnitud que se le dé a las fuerzas atractivas/repulsivas.
- Problema de mínimos locales cuando el móvil se encuentra en una zona donde no existe salida, como se puede ver en la Figura 2-14.

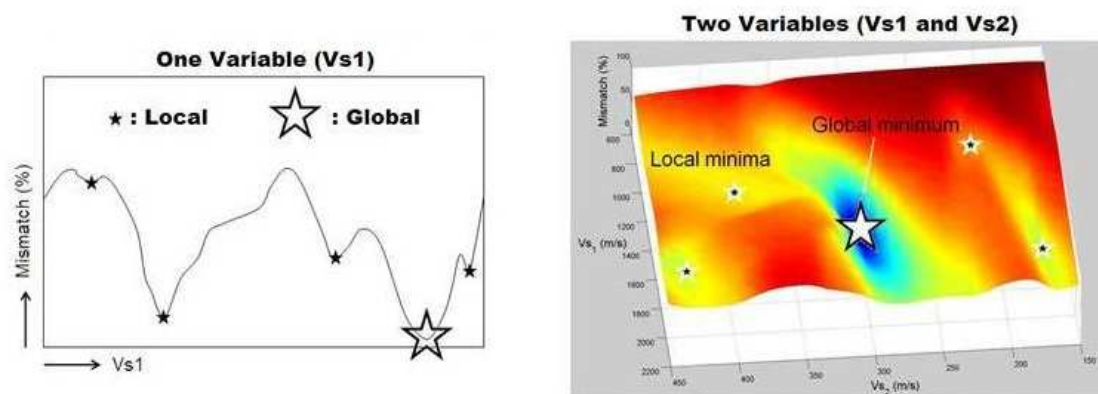
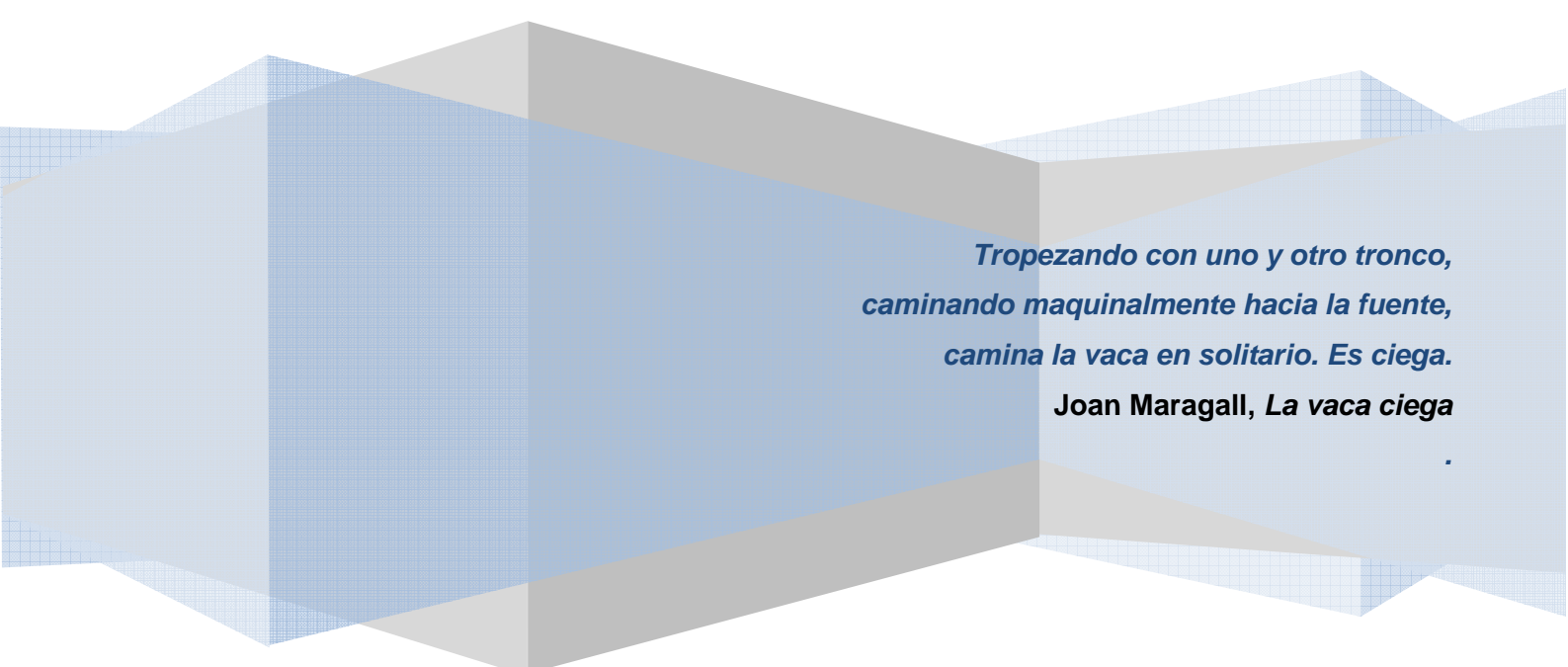


Figura 2-14: mínimos global y local [Nil09]

CAPÍTULO 3.

ENTORNO DE DESARROLLO



*Tropezando con uno y otro tronco,
caminando maquinalmente hacia la fuente,
camina la vaca en solitario. Es ciega.*

Joan Maragall, *La vaca ciega*

3. Entorno de desarrollo

Este capítulo presenta las características del entorno de trabajo, especificando la base tecnológica empleada para el desarrollo de la navegación autónoma. Cada uno de los sistemas que forman parte del vehículo van a presentar particularidades de distinta índole (técnica, logística, odométrica...) que serán necesarias ajustar en la medida de lo posible dependiendo de su magnitud e influencia en el resultado final.

De esta forma, se van a ir desarrollando los distintos aspectos previos al modelado del entorno, partiendo del prototipo de robot móvil iCab. Se realizará la descripción de las características de dicho robot, para posteriormente describir las implementaciones realizadas sobre el mismo a distintos niveles.

3.1 Arquitectura hardware

El punto de partida es el robot móvil iCab (Intelligent Campus Automobile), vehículo autónomo para exteriores, desarrollado por el Laboratorio de Sistemas Inteligentes (LSI) de la Universidad Carlos III de Madrid. La plataforma sobre la que se sustenta consiste en un vehículo eléctrico marca EZGO (Figura 3-1), que ofrece el espacio físico necesario para instalar cómodamente los ordenadores de a bordo y demás dispositivos necesarios, con facilidad en su maniobrabilidad y conducción, y manteniendo la ventaja de un volumen reducido en comparación con otros medios de transporte.



Figura 3-1: vehículo EZGO

Sobre dicha plataforma fueron necesarias una serie de transformaciones lógicas y mecánicas para dotarlo de un sistema de control electrónico. Esta tarea fue realizada dentro del contexto del Máster Oficial en Robótica y Automatización impartido por el Departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III de Madrid. El trabajo, de gran esfuerzo y dedicación, ha sido desarrollado por Pedro Portalatín Ginés y Juan Manuel Hidalgo Fiestas, y presentado como Tesis de dicho Máster [Por09] [Hid09].

Los cambios realizados se orientan a la automatización tanto de la tracción (que controla el desplazamiento del vehículo) como de la dirección (que controla la orientación del vehículo en sus desplazamientos), permitiendo su gobierno desde el ordenador encargado de la navegación. El sistema de control recibe las consignas de movimiento provenientes el ordenador responsable de generar la ruta, y se encarga de la ejecución a bajo nivel de dichas consignas. También implementa los mecanismos necesarios para chequear la integridad del hardware y producir una parada de emergencia en caso de que se detecte un fallo.

Además de poder ser controlado desde un ordenador, permite la operación manual, con el fin de facilitar la movilidad del vehículo en las tareas de desarrollo, depuración e integración de los distintos sistemas que embarca.

Una vez definidos los cambios mecánicos efectuados en el vehículo EZGO, a continuación se pasarán a comentar los distintos módulos que conforman la arquitectura del iCab [Gar09-obs], y cuyo esquema general se muestra en la Figura 3-2.



Figura 3-2: esquema hardware del equipo

3.1.1 Telémetro láser

El telémetro láser es el modelo SICK LMS 291-S05 (Figura 3-3), capaz de leer un plano de 2 dimensiones con una resolución desde 1° hasta 0.25° , y con un campo de visión configurable hasta 180° . La distancia máxima es regulable desde los 8m hasta los 80m. El periodo mínimo de muestreo del láser es de 13ms.



Figura 3-3: láser SICK LMS 291-S05

El láser fue montado en la parte frontal del coche, mediante un anclaje que provee el fabricante y adaptado para el espacio disponible en el EZGO (Figura 3-4).



Figura 3-4: estructura de soporte y láser montado en el vehículo

3.1.2 Adquisición y Procesado

El láser se comunica con un PC, dotado de una tarjeta de adquisición modelo: *DSCLP-200/300 2 port RS-422/485 Low Profile UPCI board*, de la marca *Quatech* (Figura 3-5).



Figura 3-5: tarjetas de adquisición

Este modelo permite conectarse al láser a la máxima velocidad de conexión que puede dar la el puerto RS 422 del láser, 500kbps. Pese a que la configuración de la velocidad del puerto no sigue el estándar de velocidades, fue posible la conexión a la máxima velocidad que resultó en una lectura de un escaneo cada 52msecs, con una frecuencia mayor de 19 Hz. La tarjeta provee de dos puertos, por lo tanto, es posible conectar hasta dos telémetros láser al mismo tiempo.

3.1.3 Alimentación

El sistema de alimentación está compuesto por una batería de automóvil, de 24V y de un sistema transformador de corriente alterna a continua de 1000W de potencia, unidos mediante un relé. El sistema permite alimentar el resto de sistemas mediante una batería cuando el coche se encuentra en movimiento. El inversor de corriente escogido, de 1000W es un sistema sobredimensionado, debido a que se estima que en un futuro tanto el número de equipos, y sistemas de adquisición sean mayores.

Consumo de un ordenador más pantalla: aprox. 400W.

Consumo del sistema Telémetro Láser: 20W.

3.1.4 Visualización y E/S

La visualización del sistema, se realiza mediante una pantalla táctil, modelo Xenarc 7" 700 TSV (Figura 3-6). Que permite visualizar los resultados, además de servir, junto con un teclado, como sistema de entrada de datos del ordenador.

La idea de emplear una pantalla táctil ha sido incluida en la plataforma para permitir servir de interfaz con el usuario al sistema de conducción autónoma, permitiendo elegir el destino deseado pulsando la pantalla táctil.



Figura 3-6: pantalla táctil

3.2 Arquitectura software

Para el desarrollo del sistema se emplea un ordenador ubicado en el propio vehículo. El sistema operativo empleado es Windows XP y el entorno de desarrollo Visual Studio 2008; el lenguaje de programación empleado en todas las modelizaciones realizadas es C++. La elección de estas características se debe fundamentalmente a que es preciso emplear el mismo tipo de plataforma que otras aplicaciones del grupo, para favorecer así la compatibilidad entre los distintos componentes y la portabilidad del código.

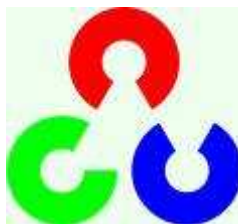
El empleo de C++ permite a su vez la integración de bibliotecas ya existentes, al encontrarse muchas de ellas implementadas en dicho lenguaje de programación.

El desarrollo del sistema de navegación precisa de la integración en el mismo de distintas bibliotecas, las cuales se exponen a continuación.



MRPT (*Mobile Robot Programming Toolkit*): librería de robótica con implementaciones a distintos niveles (modelado, análisis, planificación...). Para su compilación, se necesita instalar previamente una serie de bibliotecas, algunas obligatorias y otras opcionales (dependiendo de las funcionalidades utilizadas): entre las primeras se encuentra wxWidgets y OpenCV (en su versión 1.0.0 ó 1.1.0); mientras que para ciertas aplicaciones relacionadas con los mapas de rejilla, es necesario la instalación de las librerías Boost.

En el capítulo 6.1 se explicará con más detalle su funcionamiento y empleo en el presente proyecto.



OpenCV: biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció se ha utilizado en infinidad de aplicaciones, desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Su publicación bajo licencia BSD permite que sea usada libremente para propósitos comerciales y de investigación.

Open CV es multiplataforma, en código C y C++, existiendo versiones para Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de Visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.



Boost: es un conjunto de librerías de código abierto y revisión por pares preparadas para extender las capacidades del lenguaje de programación C++. Su licencia permite que sea utilizada en cualquier tipo de proyectos, ya sean comerciales o no.

Su diseño e implementación permiten que sea utilizada en un amplio espectro de aplicaciones y plataformas. Abarca desde librerías de propósito general hasta abstracciones del sistema operativo. Con el objetivo de alcanzar el mayor rendimiento y flexibilidad se hace un uso intensivo de plantillas. Boost ha representado una fuente de trabajo e investigación en programación genérica y metaprogramación en C++.



wxWidgets: bibliotecas multiplataforma y libres, para el desarrollo de interfaces gráficas programadas en lenguaje C++. Están publicadas bajo una licencia LGPL, similar a la GPL con la excepción de que el código binario producido por el usuario a partir de ellas, puede ser propietario, permitiendo desarrollar aplicaciones empresariales sin coste.

wxWidgets proporcionan una interfaz gráfica basada en las bibliotecas ya existentes en el sistema (nativas), con lo que se integran de forma óptima y resultan

muy portables entre distintos sistemas operativos. Están disponibles para Windows, MacOS, GTK+, Motif, OpenVMS y OS/2.

También pueden ser utilizadas desde otros lenguajes de programación, aparte del C++: Java, Javascript, Perl, Python, Smalltalk, Ruby .

3.3 Sistemas implementados o en desarrollo

A continuación se exponen los sistemas implementados o en desarrollo correspondientes al proyecto iCab [Gar09-obs].

iCab1 **LSI** Laboratorio de Sistemas Inteligentes **iCab2**

Sistema inteligente de transporte (SIT)

Sistema Laser
La medida de distancias en el entorno próximo al iCab es obtenida mediante el empleo de un láser SICK.

Sistema GPS
Sistema GPS diferencial, modelo MTI-G del fabricante Xsens. Utilizado para el posicionamiento global. Posee sensores inerciales.

Seta de emergencia inalámbrica
Parada de emergencia de la plataforma.

Movimiento Wireless
Movimiento del vehículo por medio de un mando de la Wii.

Pantalla táctil
Xenarc 7''
Dispositivo para intercambio de información con el PC.

Visión estereoscópica
Sistema de visión para el cálculo de distancias.

Comunicaciones Wireless entre iCab's
Se establecen sistemas de comunicaciones del tipo V2V (Vehicle to vehicle) V2I (Vehicle to infrastructure)

iCab2

Figura 3-7: sistemas implementados o en desarrollo

Seta de emergencia

La naturaleza del proyecto iCab requiere un especial cuidado en cuanto a elementos de seguridad, debido a la necesidad de transporte de personas, y que éste se realice de forma autónoma. Para evitar situaciones peligrosas referentes al manejo automático del sistema, se decidió implementar un sistema de seguridad a distancia, basado en un sistema transmisor/receptor. El sistema fue diseñado paralelamente a la seta de emergencia que se instaló inicialmente en la plataforma, de tal forma que ésta pueda estar o no activa, dependiendo de si el sistema estaba en modo autónomo o no.



Figura 3-8: sistema de seguridad a distancia

El sistema está diseñado para evitar errores por pérdida de señal, de tal forma que si la seta está activada, para evitar la señal de emergencia es necesario tener pulsado el botón del emisor. Gracias a esta configuración, el automóvil se detendrá, no solo ante una petición de parada de emergencia, sino que en caso de pérdida de señal, por interferencia o excesiva distancia, el sistema también daría error y se pararía.

Interfaz táctil

Se incorporó una pantalla táctil, de la marca Xenarc de 7'. Ésa pantalla pretende, en un futuro, no solo ser la interfaz de salida del coche, sino además proveer de una fuente de información al sistema. De ésta forma el usuario podría, de una forma intuitiva, decir a qué parte de un mapa virtual desearía ir.



Figura 3-9: interfaz táctil

Manejo a distancia a través de la web

Uno de los sistemas que se pretenden desarrollar para el manejo del ICab, una vez implementado el movimiento automático, es un sistema de llamada y avisos a través de la web. De tal forma que el coche reciba peticiones vía web de transporte, y éste sea capaz de proceder con el transporte de forma totalmente automática. Éste sistema sentaría las bases para un sistema completo de transporte, basado en una flota de iCabs, con gestión centralizada.

Sistema Láser

El radar láser incorporado en el automóvil es una de las herramientas más versátiles que dispone, debido a que, además de para el cálculo de trayectorias y movimientos, éste es utilizado para la detección de obstáculos típicos en entornos viarios, como peatones y otros automóviles [Gar09-lidar] [Gar09-veh]. Estos sistemas han sido desarrollados y están disponibles para ser incorporados a la plataforma.

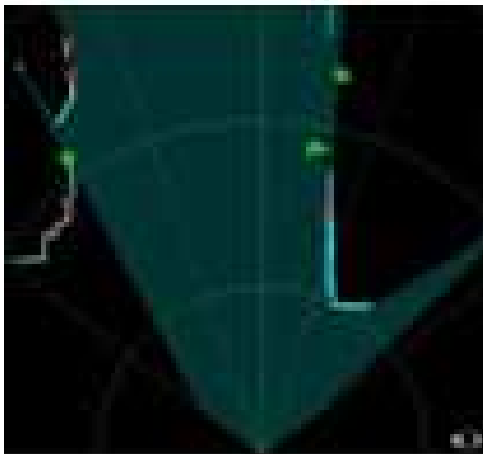


Figura 3-10: sistema láser

Visión estereoscópica

Mediante el sistema de visión en estéreo se pretende mejorar la detección de coches y peatones disponibles de detección mediante radar láser. La información que proveen las cámaras es mayor que la disponible por el láser, pero a costa de una menor fiabilidad. De ésta forma, combinando ambos sensores se puede dar una estimación bastante razonable de la detección hecha.



Figura 3-11: visión estereoscópica

Comunicación V2V (Vehicle-Vehicle) y V2I (Vehicle-Interface)

La comunicación vehículo-vehículo, y vehículo-infraestructura es uno de los campos de investigación con mayor interés en el panorama científico internacional. Añadiendo una segunda plataforma iCab, se pretende realizar investigaciones en éste ámbito, siendo algunos de los sistemas propuestos los correspondientes al intercambio de información de sensores, información a alto nivel (obstáculos, personas...), y el manejo inteligente y automático de intersecciones.

Manejo inalámbrico mediante mando a distancia (Wii)

Para posibilitar la teleoperación, y con el fin de probar diferentes protocolos de comunicación, se desarrolló un sistema de control remoto, basado en un mando inalámbrico de la videoconsola Wii. Hay que destacar que ésta comunicación es de alto nivel; es decir, es el mando quien se comunica con el ordenador de abordo instalado en la plataforma, y éste, en modo automático, envía las órdenes de movimiento al coche.

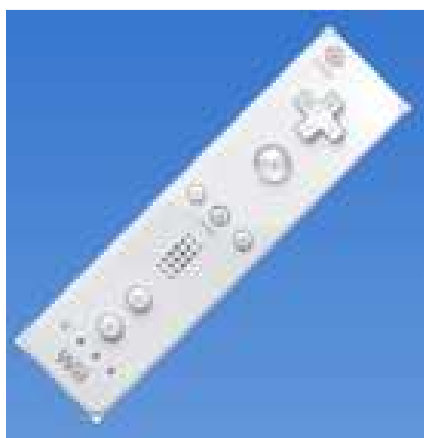


Figura 3-12: mando inalámbrico (Wii)

Sistema GPS para corrección del movimiento del coche

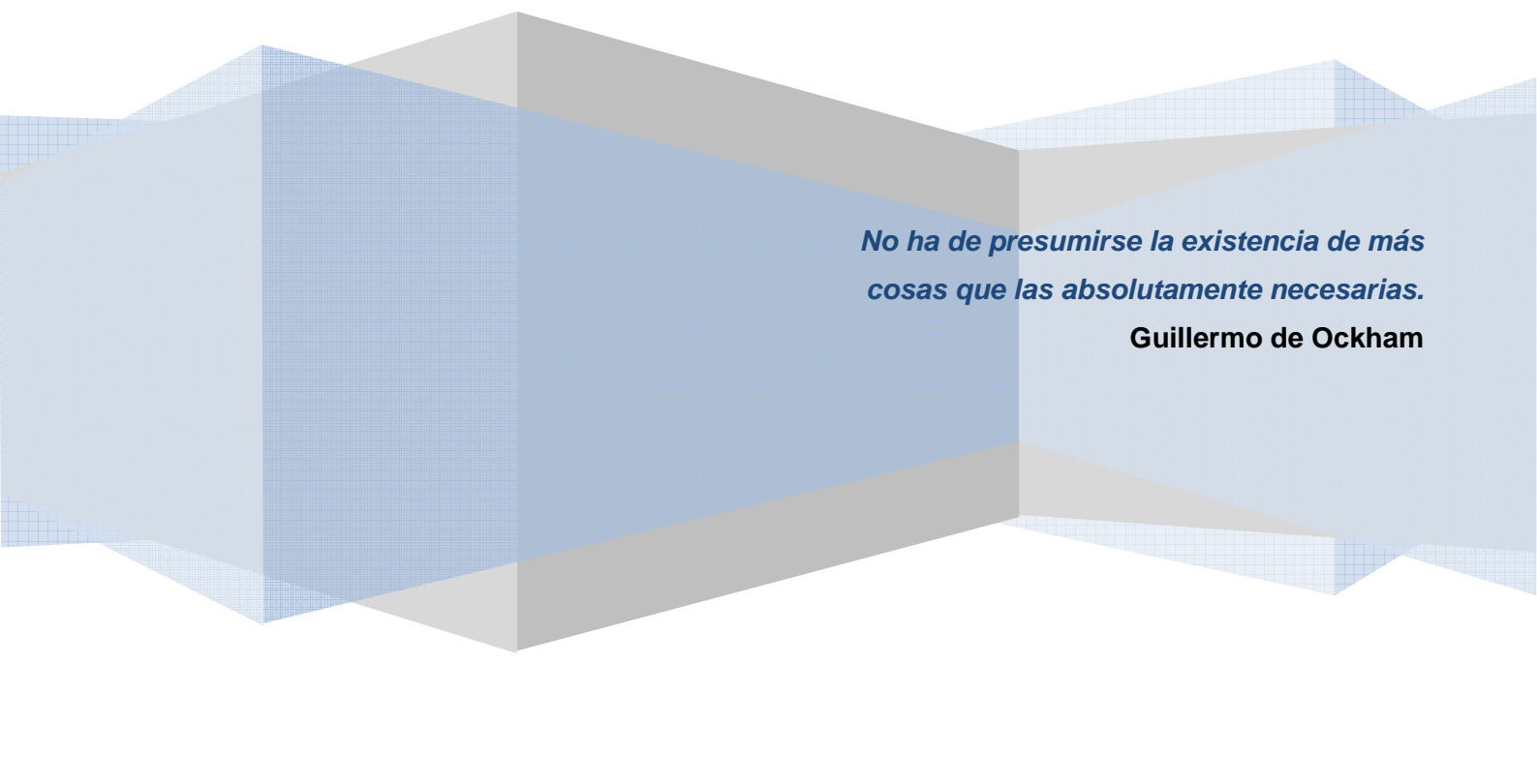
El sistema GPS, empleado para la solución del problema del presente proyecto. Fue previamente incorporado a la plataforma para solventar otros problemas, como son aquellos relativos a la detección de peatones y coches, y al intercambio de datos entre vehículos.



Figura 3-13: dispositivo GPS

CAPÍTULO 4.

PERCEPCIÓN DEL ENTORNO FÍSICO



*No ha de presumirse la existencia de más
cosas que las absolutamente necesarias.*

Guillermo de Ockham

4. Percepción del entorno físico

Pongamos como ejemplo el conocimiento humano aplicado a una situación similar. Una persona quiere recorrer el camino que le lleva, desde su situación actual, a un destino conocido. Si ya ha recorrido dicho camino anteriormente, tendrá una idea general del recorrido por donde debe ir. Mentalmente, podrá trazar la ruta, distinguiendo en ella dónde se sitúan los edificios y muros, por dónde se hayan las vías de tránsito, y sabrá cuándo debe girar a izquierda o derecha y cuándo seguir recto. Si alguien le pregunta la ruta, podrá orientarle mediante una aproximación del camino a seguir (*“siga recto, tuerza la primera a la derecha, y cuando haya pasado el quiosco de helados, ya estará usted en el parque”*)

Sin embargo, lo más usual es que no recuerde ciertos detalles más específicos, o que omita elementos no determinantes para el recorrido, como por ejemplo no indicará que existen bancos que deben ser rodeados, o que debe pasar entre dos estatuas (*“siga recto, rodee un árbol, no se choque con las papeleras, rodee otro árbol pero más grande que el primero, tuerza la primera a la derecha, pase una fuente, otro árbol, dos bancos colocados frente a frente, y cuando haya pasado el quiosco de helados, ya estará usted en el parque”*)

Y desde luego, no podrá proporcionar información sobre las variaciones experimentadas de forma permanente por el entorno pero que aún no conoce (*“pero puede ser que el segundo árbol- el que es más grande que el primero-lo hayan talado, así que hay posibilidades de que no tenga que preocuparse por él”*) o sobre los cambios dinámicos en el mismo (*“y tenga cuidado de no chocarse con la gente!”*)

Por ello, el robot móvil, aunque orientado hacia dónde debe dirigirse, necesita una visión propia del medio en el que se encuentra que evite la colisión con los obstáculos fijos o en movimiento que encuentre a su paso, así como que permita saber de antemano si es factible alcanzar las submetas ya marcadas a las que debe dirigirse (y que pueden estar bloqueadas o ser temporalmente inaccesibles).

4.1 Particularización del entorno

El robot se encuentra situado en un entorno exterior del que posee información general, como es el patio de la Universidad. Para simplificar el modelo, se ha optado por reducir las dimensiones del espacio, concretándolo a la parte correspondiente a la Figura 4-1.



Figura 4-1: plano aéreo de la Universidad Carlos III

De dicho entorno se conocen a priori la situación de los edificios Betancourt y Sabatini, los elementos delimitantes (muro que separa la parte posterior al edificio Betancourt de la calle), algunos elementos estáticos de mayor tamaño (como la torre de refrigeración) y aquello que impide el paso del vehículo y que no podrá ser detectado a posteriori debido a su posición y/o dimensiones (pivotes del suelo que impiden el paso a vehículos).

Partiendo de un modelo de robot móvil en 2 dimensiones, la zona donde se desarrollarán las pruebas será una superficie plana, sin cambios bruscos en la altura del terreno.

4.2 Sistema de percepción

El sistema de percepción proporciona al robot la capacidad para efectuar movimientos eficientes en un entorno no conocido y variable. Aunque se asumen unas condiciones iniciales en cuanto a la situación y topología donde se desarrollarán las pruebas, sólo se dispondrá previamente de información global del medio, sin reunir las características topológicas próximas al móvil. Por ello se hace necesario implementar un modelo del entorno que permita al robot evaluar las condiciones cambiantes del medio y sus propias acciones sobre el mismo, permitiéndole la modificación de estas últimas en caso necesario.

Debido a la complejidad requerida en el proceso de percepción, dicha tarea puede separarse en tres fases de desarrollo [DEI]:

- Adquisición de la información. La información sobre el medio situado en un entorno cercano al robot puede ser obtenida mediante el empleo de diversos tipos de sensores, que se pueden clasificar de forma básica en dos grupos: sensores de distancia y sensores de contacto. En la implementación propuesta, dadas las características del medio y las condiciones de navegación, se hace necesario el uso de dispositivos que permitan obtener la distancia a la que se encuentran los obstáculos y, por consiguiente, aporten información sobre el espacio libre donde puede transitar el móvil. Dicha información es obtenida mediante telemetría láser (apartado 4.2.1).
- Tratamiento de los datos. Su objetivo principal radica en el procesado de la información proporcionada por la fase anterior, por lo que su funcionamiento depende en gran medida de la cantidad y calidad de la misma. Abarca aspectos diversos, como la corrección debido a distorsión o ruido, la clasificación de la información útil o representativa, y el modelado

de la misma para la extracción de patrones básicos. La aplicación de esta fase en el presente proyecto se orienta a buscar una relación en el conjunto de datos independientes que permita su agrupación para formar estructuras lineales (segmentos), como se desarrolla en el apartado 4.2.2.

- Interpretación del entorno. Corresponde a la obtención de un modelo que se asemeje adecuadamente la realidad, mediante el empleo de técnicas de inteligencia artificial y reconocimiento de formas. El resultado de esta fase será una reconstrucción lo más precisa posible del entorno, o dicho de otra forma, de los obstáculos que caracterizan el mismo. Partiendo del conjunto de segmentos que representan los contornos medidos, se realiza un análisis de la situación actual, comparándola con la información previa almacenada y tomando decisiones sobre su validez (apartado 4.2.3).

4.2.1 Fase de adquisición - dispositivo láser

La medida de distancias en el entorno próximo al robot es obtenida mediante el empleo de un láser SICK modelo LMS 291-S05 (Figura 4-2) [SICK]. Este tipo de dispositivos es utilizado en un amplio rango de aplicaciones: monitorización de áreas, medidas y detección de objetos, prevención de colisiones y posicionamiento, entre otros. Pertenece a la categoría de sistemas de medida sin contacto directo (*non-contact measurement systems, NCSDs*). Su principio de funcionamiento es simple: realiza un escáner óptico del entorno en tiempo real y de forma bidimensional (en función del radio de apertura del haz láser). Los objetos a medir no requieren características especiales de detección (reflectores, marcas o algún tipo de iluminación especial), lo que lo convierte en un dispositivo versátil.



Figura 4-2: láser SICK LMS 291-S05

El mecanismo de medida se basa en el tiempo de vuelo de los pulsos de luz del láser: los rayos son emitidos y reflejados si encuentran un objeto, siendo registrados posteriormente por el receptor (Figura 4-3). El tiempo entre la transmisión y la recepción del impulso (tiempo de vuelo) es directamente proporcional a la distancia entre el dispositivo y el objeto.

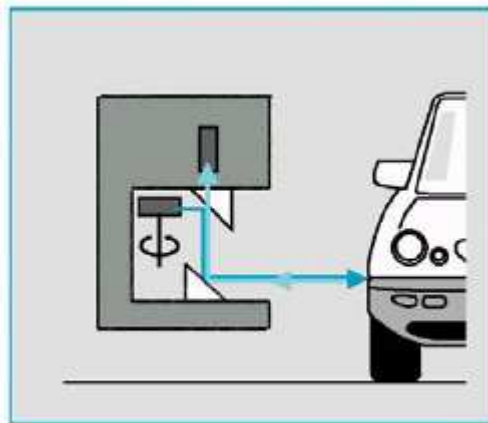


Figura 4-3: rayos emitido y reflejado

El rayo láser es desviado por un espejo de giro interno, de modo que se realiza una exploración en forma de abanico del área circundante (Figura 4-4). El contorno

del objeto medido es determinado a partir de la secuencia de impulsos recibidos. Los datos de medida están disponibles en tiempo real para su posterior evaluación a través de la interfaz serie.

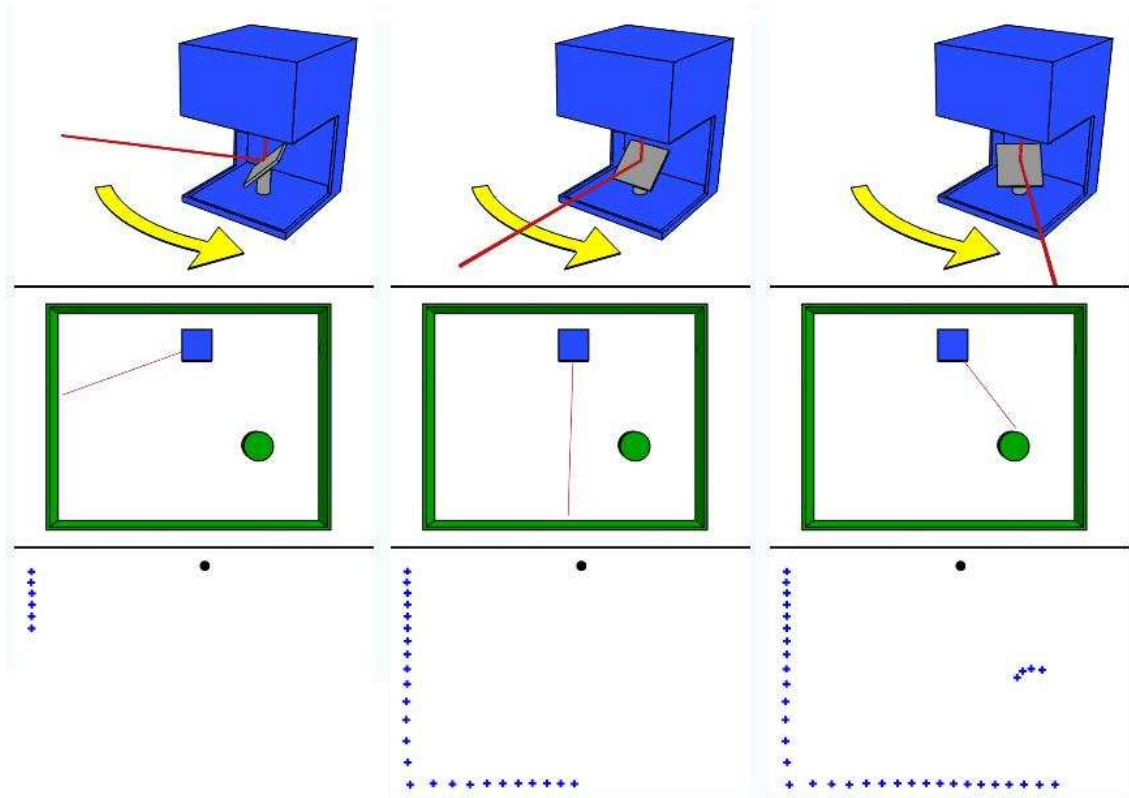


Figura 4-4: funcionamiento del espejo interno del láser [LIDAR]

Las características del dispositivo permiten distintas configuraciones del mismo, mediante la variación de algunos parámetros como son: el campo de visión o margen de exploración (*Field of view*), de 90° a 180° ; la máxima distancia de alcance de los haces (*Operating range*), de 0m a 80m; y la resolución angular o grados de separación entre los haces (*Angular resolution*), de 0.25° a 1° . En la Tabla 4-1 se muestra un resumen de las propiedades del láser, incluidas las ya mencionadas.

Features

Type :	Mid-Range
Field of application :	Outdoor
Light source :	Infrared (905 nm)
Laser class :	1 (EN/IEC 60826-1), Eye-safe
Field of view :	90 °
Scanning frequency :	75 Hz
Operating range :	0 m ... 80 m
Max. range with 10 % reflectivity :	30 m
Angular resolution :	0.5 °
Heating :	Optional via external heating plate
Fog correction :	yes
MTBF :	50.000 h
Resolution :	1 mm

Performance

Detectable object shape :	Almost any
Systematic error :	± 35 mm
Statistical error :	10 mm

Interfaces

Serial (RS-232, RS-422) :	✓
Ethernet :	-
CAN-Bus :	-
PROFIBUS :	-
PROFINET :	-
DeviceNet :	-
Switching inputs :	1
Switching outputs :	0
Optical indicators :	3 LED

Mechanics/electronics

Electrical connection :	2 Interface plug with 9 pin D-Sub socket (solder connection)
Operating voltage :	24 V DC ± 15 %
Power consumption :	30 W
Housing colour :	Light gray (RAL 5012)
Enclosure rating :	IP 65
Protection class :	2, insulated
Weight :	4.5 kg
Dimensions :	156 mm x 155 mm x 210 mm
Housing :	Aluminium die-cast

Ambient data

Object remission :	1.8 % ... 10,000 %
Electromagnetic compatibility (EMC) :	EN 61000-6-2 / EN 61000-6-3 / A11 (2004-07)
Vibration resistance :	IEC 68
Shock resistance :	IEC 68
Ambient operating temperature :	0 °C ... 50 °C
Storage temperature :	-30 °C ... 70 °C, with heating plate -12 ... +50 °C

Tabla 4-1: características del láser SICK LMS 291-S05

La configuración seleccionada en el presente proyecto permite barrer un espacio de 90° a cada lado del eje central del láser; en otras palabras, proporciona un ángulo de barrido de 0° a 180° , encontrándose el eje central del dispositivo en 90° (Figura 4-5). La obtención de información la realiza mediante la resolución angular, cuyo valor es puesto a 1° , de forma que obtendremos un dato por cada uno de los ángulos inspeccionados; esto es, tendremos 181 valores de distancia a objetos.

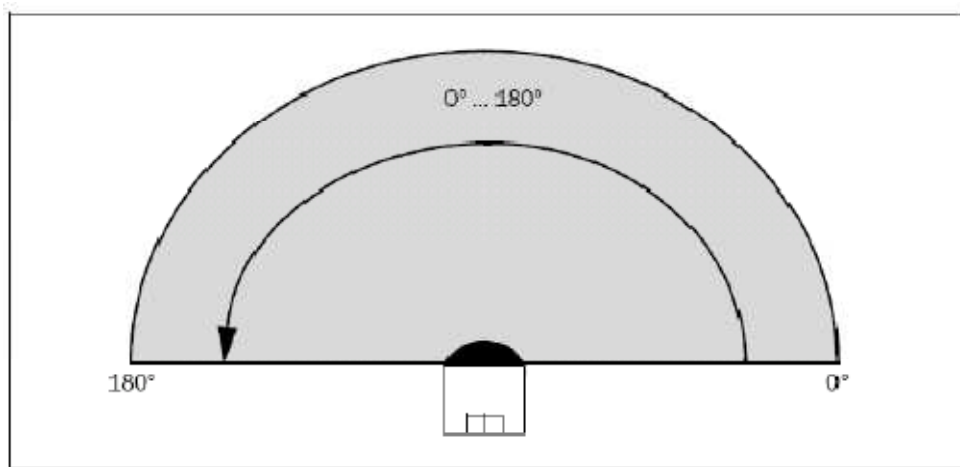


Figura 4-5: ángulo de barrido del láser

El alcance máximo del láser también es un parámetro configurable, y es fijado a – aproximadamente - 80 metros. Esta distancia, aunque superior a la necesaria para un modelado próximo al objeto móvil, aportará una perspectiva más amplia del entorno.

Los datos suministrados por el láser ofrecen información de los obstáculos de la siguiente manera: si la distancia al objeto es inferior al alcance, denota la existencia de un cuerpo sobre el que se ha producido la reflexión del haz correspondiente y, por tanto, la existencia de un obstáculo en esa posición; si la distancia se aproxima al alcance, no se ha encontrado nada que intercepte el haz y por tanto, se trata de posiciones sin obstáculos en principio.

La visualización de los datos obtenidos es un proceso que se puede realizar de forma instantánea una vez se han obtenido los valores del entorno; el procesamiento de los mismos y su tratamiento para construir el conjunto de obstáculos es un proceso más amplio, que requiere una lógica adicional, y será analizado posteriormente en los capítulos 4.2.2 y 4.2.3.

Problemas derivados del láser

El láser se presenta como un dispositivo con un margen de exactitud adecuado para la toma de medidas en espacio libre, permitiendo obtener información de forma precisa sobre las distancias a los objetos sin necesidad de complicados tratamientos previos de los datos obtenidos. Sin embargo, no se encuentra exento de complicaciones.

La principal de ellas, más que en proceder del propio mecanismo de detección, deriva de las superficies externas donde se produce la reflexión de la señal. De están forma, aquellos materiales que absorban luz, como pueden ser los negros intensos, no serán detectados por el láser y se considerará el espacio que ocupan como una zona libre de tránsito, cuando no lo es. Otro problema radica en el ángulo de incidencia sobre los cuerpos, que puede dificultar la correcta detección aun tratándose de superficies reflectantes.

Por otra parte, no sólo va a existir una pérdida de información, sino también la aparición puntual de datos incorrectos que pueden alterar el modelo de entorno, relacionados con ciertas situaciones ambientales (como es el caso de humo, niebla, o partículas en suspensión).

Al tratarse estas de situaciones externas al elemento de medida, la dificultad de identificar los errores se ve incrementada por la falta de conocimiento a priori del entorno local próximo al móvil. De forma lógica, no resulta adecuado añadir más información que la existente (y detectada) en cada momento; pero sí es posible identificar los datos obtenidos de forma puntual (sin relación con detecciones anteriores), o que hayan desaparecido (tras haber sido detectados previamente). Estas situaciones serán tratadas por el sistema de modelación de obstáculos que, como se ha indicado anteriormente, podrá verse en el capítulo 4.2.3.

4.2.2 Fase de tratamiento - clasificación de la información

Llegados a este punto, el sistema posee un conjunto de datos que representa las distancias obtenidas en el entorno del móvil, una para cada ángulo de detección. Ahora es necesario realizar el tratamiento de esta información para construir con ellas estructuras útiles con las que sea posible interactuar.

Las distancias medidas van a seguir un cierto patrón en caso de estar asociadas al mismo contorno de obstáculo. De esta forma, es posible agrupar puntos cercanos como pertenecientes al mismo objeto, y reducir la información relevante a un conjunto de segmentos.

El análisis de las muestras comienza con un proceso de ajuste en función de ciertos parámetros configurables, que permiten una búsqueda más o menos refinada de patrones. Cada elemento identificado se asocia a los segmentos que lo representan; este proceso implica pérdida de información, al aproximar cualquier forma de contorno a un conjunto de rectas. La clasificación de los distintos tipos obtenidos permite además añadir conocimientos sobre la naturaleza de los mismos.

Tras el proceso de tratamiento de la información, el conjunto resultante se puede emplear directamente como representación del espacio sin necesidad de lógica adicional. Sin embargo, debido al carácter dinámico del entorno (tanto por los cambios dentro del mismo, como debido a las variaciones de percepción provocadas por el movimiento del vehículo), se hace necesario ampliar este modelo para proporcionar un significado a la simbología obtenida, así como para buscar las relaciones entre los esquemas de datos adquiridos a medida que se van obteniendo nuevos datos y actualizando los mapas. Este proceso es contemplado a continuación en el capítulo 4.2.3.

4.2.3 Fase de interpretación - modelado de obstáculos

Una vez se dispone del conjunto de las formas que rodean al robot, el sistema debe ser capaz de proporcionar un significado al conjunto de símbolos y construir a partir de ellos el mapa de la realidad observada. Para ello, debe contar con un mecanismo

que le permita evaluar las condiciones actuales en contraposición con las ya adquiridas, de forma que sea capaz de razonar si las medidas obtenidas en dos instantes distintos de tiempo corresponden a un mismo objeto.

Los segmentos proporcionados por la fase anterior reciben a partir de este momento el tratamiento de obstáculos. En un primer momento, se planteó la posibilidad de modelarlos de dos maneras diferentes: como contornos o como polígonos rectangulares. Ambas opciones fueron implementadas en la práctica para evaluar sus prestaciones.

El modelado mediante polígonos permitía crear obstáculos asociando a cada segmento un área (una cierta cantidad de celdas); ofrecía el interés de trabajar con zonas más amplias del espacio, incrementándose la ocupación del medio. El principio de funcionamiento era el siguiente: a partir de las formas lineales obtenidas en la fase previa de tratamiento, se ampliaba el espacio que rodeaba a dichos segmentos hasta un valor fijo, asociando dicho área al obstáculo.

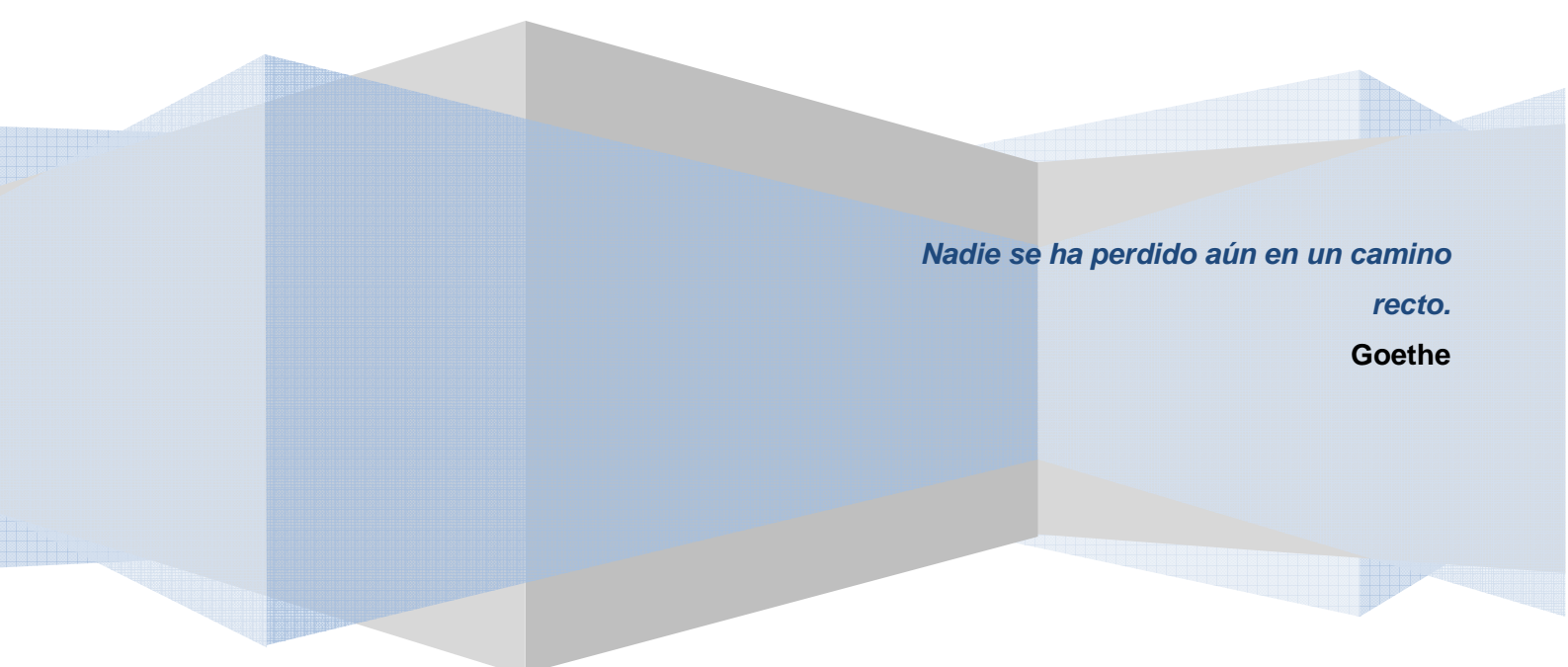
Sin embargo, se desechó esta idea, ya que resultaba complicado ajustar qué zonas/celdas correspondían a cada obstáculo en cada momento (sobre todo, cuando se producía un cambio en la orientación del robot), aumentándose el tiempo de ejecución y la carga computacional (ya que, tanto en la construcción como en la comparación de obstáculos, era necesario trabajar con las celdas de una en una). Finalmente, se optó por emplear los contornos de obstáculos como medida de representación.

Los obstáculos pertenecientes a cada muestreo del espacio son comparados con los anteriormente obtenidos, buscándose algún tipo de relación que permita identificar elementos similares. Los elementos considerados como próximo (y por tanto, factibles de ser identificados como el mismo obstáculo) sufrirán una serie de comprobaciones para asegurar su grado de similitud, como se desarrollará posteriormente en el capítulo 7.5. Los márgenes dentro de los cuales se puede considerar que se trata del mismo objeto es un parámetro que debe ser evaluado en función de la precisión de los sensores y de la rapidez con la que el robot realiza sus movimientos.

Los objetos ya considerados como existentes constituyen una base de información para el robot (una especie de memoria), por lo que estarían vinculados al sistema durante toda su ejecución. Para que la actualización del entorno sea correcta, es necesario realizar periódicamente algún tipo de restructuración de los obstáculos almacenados que sólo permita conservar los más significativos y actuales, y en caso de cambios bruscos en el medio percibido (por ejemplo, ante un cambio prolongado o permanente de la orientación), que permita la eliminación de toda la información almacenada.

CAPÍTULO 5.

LOCALIZACIÓN EN EL ENTORNO FÍSICO



*Nadie se ha perdido aún en un camino
recto.*
Goethe

5. Localización en el entorno físico

Junto al conocimiento del entorno (tratado en el capítulo 4), el otro gran problema que se plantea una persona a la hora de seguir una ruta cualquiera estriba en la necesidad de conocer el lugar donde se encuentra en un determinado momento. Una vez conoce la ruta y puede recrear una imagen más o menos robusta del entorno que le rodea, permitiéndole solventar las dificultades que encuentre a su paso, necesita conocer su posición en el espacio, una especie de “*se haya usted aquí*” con un círculo rojo que le indique dónde está dentro del mapa y le permita razonar a dónde debe dirigirse.

5.1 Sistema de localización

El sistema de localización permite al robot conocer su posición a medida que avanza por el medio, tanto de forma absoluta (si tomamos un sistema de referencia global, como pueden ser las coordenadas geográficas) como en relación a un punto determinado de su trayectoria. De esta forma, le permite conocer en qué parte del camino se encuentra, encaminándole a través del mismo y evitando su pérdida.

Para entender la problemática asociada a la localización, es preciso analizar los casos existentes en función de la naturaleza del entorno y del conocimiento inicial del propio robot, lo que lleva a clasificar las situaciones según tres criterios fundamentales [Thr05]:

- Localización local vs localización global. Depende del tipo de conocimiento existente al inicio y en tiempo de ejecución. El problema de ámbito local se conoce como *rastreo* (*position tracking*), y asume una posición inicial conocida; la *localización global* (*global localization*), por el contrario, parte del desconocimiento de dicha posición.
- Entorno dinámico vs entorno estático. La localización es dependiente del tipo de entorno: en el caso *estático*, el entorno es fijo y la única variable es la posición del robot; en el caso *dinámico*, la situación se complica al

producirse cambios en el medio, cuyo impacto depende de la duración y periodicidad de los mismos.

- Localización pasiva vs localización activa. Depende de la relación entre las operaciones de localización y los mecanismos de movimiento del robot. De forma *pasiva*, las acciones realizadas son independientes y no van encaminadas a facilitar la localización; mientras que de forma *activa*, existen relaciones entre ambos procesos que se orientan a ajustar la posición en la que se encuentra el robot.

En cuanto al mecanismo empleado para localizar al robot en el medio, existen diversas formas de afrontar el problema [Thr05]: separarlo como un bloque independiente ó solventarlo de forma paralela al problema de modelado del entorno. Esta segunda opción es conocida como SLAM (*Simultaneous Localization and Mapping*), y consiste en combinar las acciones de construcción del mapa y el posicionamiento del robot dentro el mismo. La localización del móvil se reduce al reconocimiento de la posición que ocupa en su entorno próximo.

Así, partiendo del modelo del espacio que rodea al vehículo, y conociendo ciertas características topológicas globales del medio, se puede hallar su situación en un momento dado. El problema se presenta en caso de no existir características discriminatorias en el trayecto que permitan relacionar el entorno local con el mapa global. Para solventar esas dificultades de asociación, se puede emplear sistemas de visión artificial y procesamiento de las imágenes obtenidas, así como sensores externos (ultrasonidos, infrarrojos y señales de radio) combinadas con balizas situadas a lo largo del recorrido.

Aún cuando las técnicas de SLAM han demostrado su eficacia en entornos exteriores, exigen ciertos requisitos como son el disponer del material técnico adecuado y respaldarse en algoritmos que modelen la información obtenida. En la implementación de este proyecto, se ha optado por la opción de solucionar el problema de localización mediante un mecanismo independiente, de forma que no sea necesario evaluar el mapa del entorno para conocer la posición.

5.2 Tecnología GPS

El mecanismo empleado para la localización del móvil es un sistema GPS diferencial, modelo MTi-G del fabricante Xsens (Figura 5-1).

Se trata de un dispositivo de medida de pequeño tamaño y bajo peso, empleado para el control y la navegación en diferentes escenarios, como aeroespaciales y automovilísticos. Consta de un sistema de posicionamiento global GPS ayudado de un MEMS (sistema microelectromecánico) basado en una Unidad de Medida Inercial (IMU) y un sensor de presión estático.



Figura 5-1: Xsens MTi-G

La IMU se compone de una serie de sensores inerciales (acelerómetros y giroscopios) y de orientación (brújula) que permiten obtener información sobre las acciones efectuadas en ese momento por el robot sin necesidad de producirse la conexión con los satélites. Dicha información está compuesta por la aceleración (en un sistema tridimensional de coordenadas cartesianas –incluyéndose la fuerza de gravedad sobre el mismo); las velocidades angulares o de giro en los tres ejes; y las fuerzas magnéticas o influencia de los campos magnéticos terrestres, con los que se obtiene la orientación respecto al Norte geográfico.

En cuanto a los datos recibidos a través de los satélites, éstos se componen de las coordenadas en las que se encuentra el aparato (latitud, longitud, altura).

El resto de información útil suministrada por el dispositivo se obtiene (dependiendo del modo de funcionamiento seleccionado) a partir del conjunto de datos procedentes de los satélites, los sensores inerciales y la brújula; tal es el caso de los ángulos de Euler que denotan la orientación del dispositivo (*roll* o alabeo, *pitch* o cabeceo, *yaw* o guiñada); y la velocidad a la que se desplaza por el medio (referenciada en el sistema tridimensional cartesiano).

El dispositivo permite seleccionar distintos modos de funcionamiento, de los cuales se pueden destacar dos: el *automotive* y el *aerospace*.

El modo *automotive* es empleado para medidas en vehículos terrestres, mientras que el *aerospace* se utiliza principalmente en sistemas de vuelo. El modo *automotive* muestra mejores resultados en movimiento que en ausencia de él, al emplear datos vía satélite para realizar los ajustes; mientras que, si el vehículo está en reposo, los datos ofrecidos cuentan con una deriva significativa. Por su parte, el modo *aerospace* sólo emplea la información GPS para calcular la posición, lo que ofrece mayor estabilidad en la misma en caso de reposo del vehículo, pero peores prestaciones en caso de movimiento (al no tenerse en cuenta el desplazamiento). En la Tabla 5-1 se muestra la influencia de las medidas obtenidas con los distintos sistemas en el cálculo de la información suministrada.

	<i>automotive</i>				<i>Aerospace</i>			
	velocidad	ang. Euler	posición	aceleración	velocidad	ang. Euler	posición	Aceleración
sensores inerciales	X	X	X	X	X	X		X
Brújula	X	X	X	X	X	X		X
GPS	X	X	X	X			X	

Tabla 5-1: sistemas empleados en el cálculo de la información (según el modo de funcionamiento)

Las necesidades del proyecto exigen conocimiento de la posición y orientación del vehículo, dentro de unos márgenes de exactitud acotados; teniendo en cuenta que el vehículo se hallará en movimiento la mayor parte del tiempo, se ha decidido emplear

el modo *automotive*, al presentar mejores prestaciones que el *aerospace* en el cálculo de la posición. Dicha posición podrá ser representada, tras el ajuste realizado gracias a los sensores inerciales y la brújula, mediante la latitud y la longitud en la que se encuentra; así como es posible modelar la orientación empleando únicamente el ángulo de guiñada, ya que en el plano bidimensional planteado sólo es necesario conocer el ángulo respecto a uno de los ejes (en este caso, empleando como referencia aquel que se corresponda al Norte geográfico)

Problemas del dispositivo GPS

Es necesario señalar la importancia de poseer un conocimiento adecuado de la localización, tanto del vehículo como de las submetas a alcanzar y de los obstáculos a lo largo del camino. Los datos recibidos vía satélite pueden no ser totalmente exactos, pero sí deben ser una aproximación lo suficientemente cercana a la realidad del vehículo como para indicar la zona en la que este se encuentra y cómo se distribuyen los obstáculos a su alrededor.

La información proporcionada por el dispositivo GPS será tan veraz como permita situar al móvil en el lugar en el que realmente se encuentra (ver Figura 5-2); por desgraciada, los datos procedentes de los satélites cuentan con factores que disminuyen la fiabilidad de los mismos. La exactitud de la información depende en gran medida del número de satélites detectados, de su posición y de la intensidad de la señal recibida de los mismos. Los datos obtenidos precisan un cierto margen temporal para estabilizarse, impidiendo la adquisición real de los mismos a medida que son reclamados. Además existe un error debido a la deriva del dispositivo, que proporciona márgenes de error aún mayores.

De esta forma, aunque algunos fallos en la medida pueden deberse al propio dispositivo (como, por ejemplo, debido a imprecisiones en el reloj interno del mismo) la mayor parte de los problemas que ocasionan la degradación de la señal GPS (y por tanto, su falta de precisión) proceden de fuentes externas, como son [OYP]:

- Retrasos de la señal al atravesar la atmósfera terrestre.
- Reflexión de la señal en objetos cercanos al receptor, lo cual incrementa el tiempo de viaje de la señal.
- Errores de órbita (también llamados errores de efemérides), que pueden llevar a fallos en la localización de los satélites.
- El número de satélites visibles por el receptor puede variar a causa de la geografía, los obstáculos o la presencia de interferencias.
- La geometría de los satélites (su posición relativa en un momento dado) resulta poco conveniente si estos se encuentran en línea o agrupados.

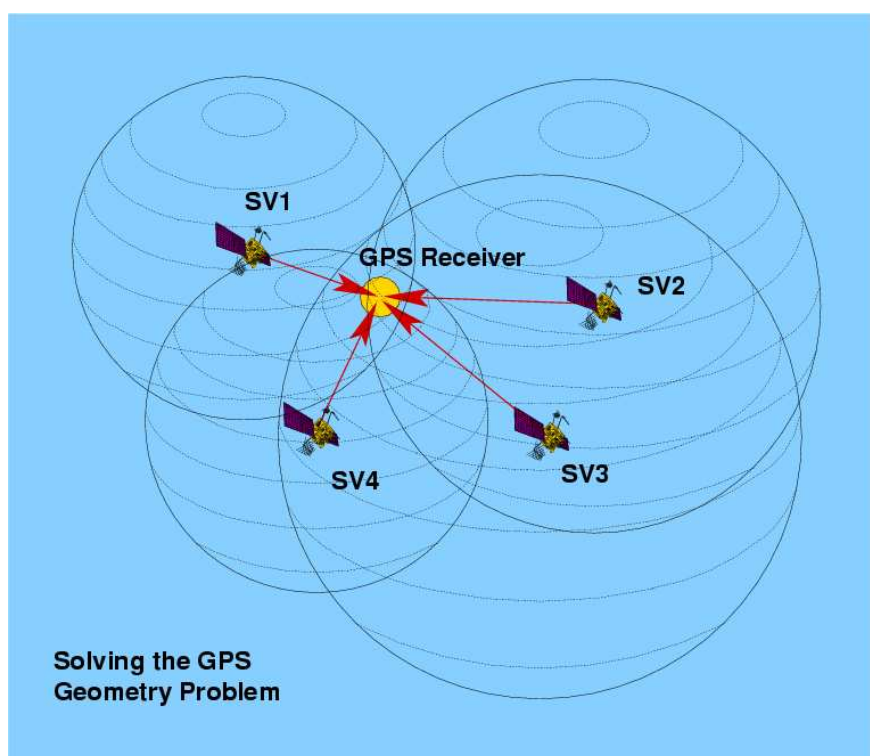


Figura 5-2: posición de los satélites para recepción de señal [AAP]

Se ha comprobado de forma experimental en este proyecto, que la fiabilidad varía en determinadas situaciones, como son:

- El momento del día en el que se toman las medidas, siendo menos fiables durante la mañana y proporcionando mayor fiabilidad a medida que avanza el día.
- Partir de una situación que se pueda considerar cercana a la realidad favorece que las medidas tomadas a partir de ese momento sean más próximas a la realidad. Si el estado inicial del vehículo presenta un error que se pueda considerar fuera del margen adecuado, los datos obtenidos a posteriori contarán con un error cada vez mayor.
- El error debido a la deriva del dispositivo parece acotarse cuando éste se encuentra en movimiento, pudiendo crecer de forma incontrolada si el móvil permanece parado. Por ello, el algoritmo de localización debe emplearse con el robot en movimiento y con el dispositivo en modo *automotive*.
- La proximidad a edificios aumenta el efecto de apantallamiento, lo que dificulta la comunicación con los satélites; es lo que se conoce como *urban canyons* (Figura 5-3). Mantenerse a una distancia prudente de los mismos – siempre que esto sea posible - facilita la obtención de datos estables.

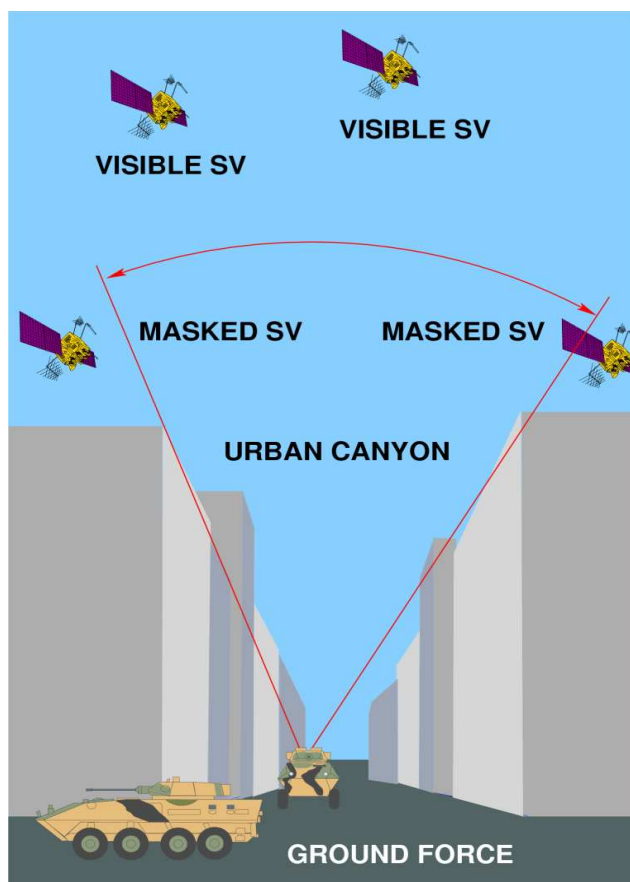
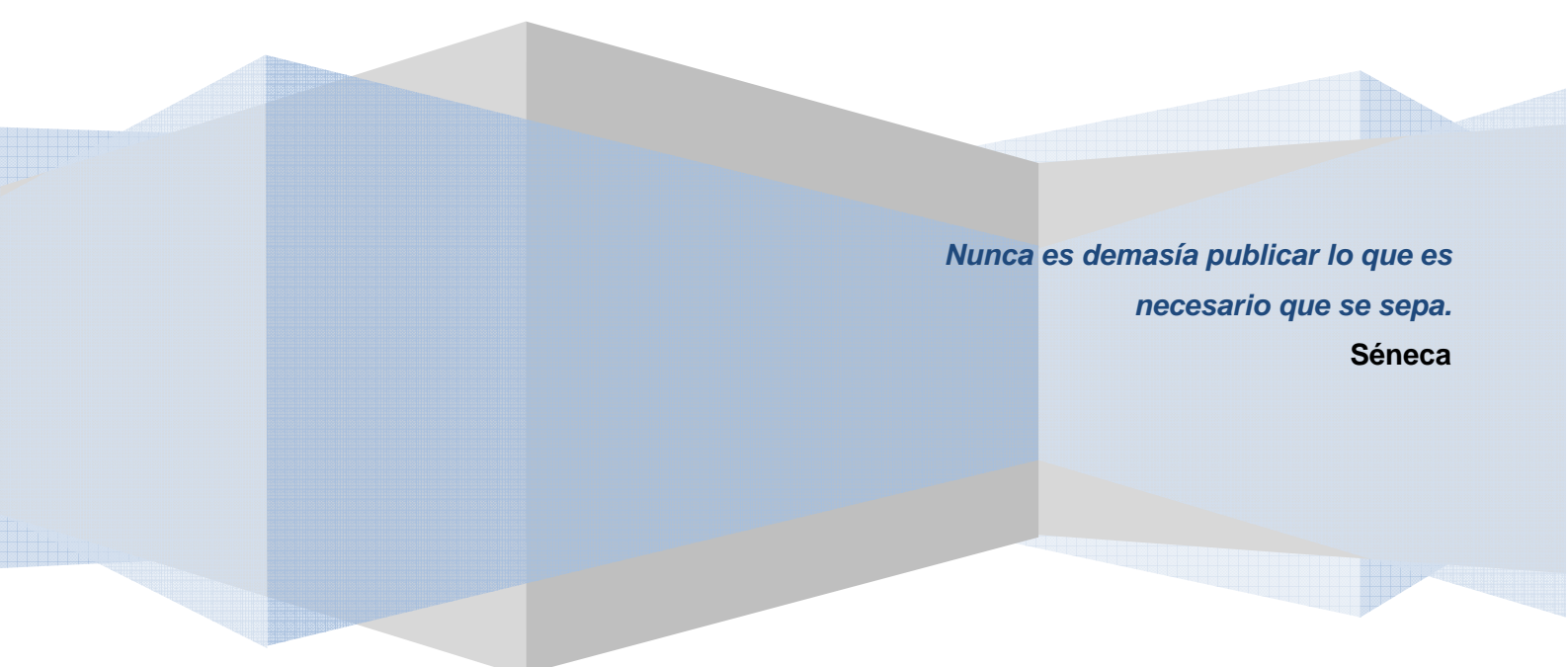


Figura 5-3: problema debido a la presencia de *urban canyons* [AAP]

A la hora de realizar las pruebas, se ha de procurar tener en cuenta los requisitos anteriores para aumentar la fiabilidad del dispositivo. No obstante, no es posible asegurar la exactitud de las muestras obtenidas, por lo que debe siempre recordarse que la posición se trata de una aproximación a su valor real, pudiendo provocar errores de localización que impidan la correcta navegación del vehículo.

CAPÍTULO 6.

ESQUEMA DE PROGRAMACIÓN



*Nunca es demasía publicar lo que es
necesario que se sepa.*

Séneca

6. Esquema de programación

El desarrollo de un sistema de navegación de la envergadura del presente proyecto conlleva una serie de necesidades de distinta índole y a distintos niveles de implementación. En este apartado se expondrán las necesidades a nivel de programación y se explicarán las características del código empleado y de los distintos módulos en los que se divide la realización del sistema.

6.1 Librería para implementaciones en robótica: MRPT

En la actualidad, existe una amplia variedad de bibliotecas de software, tanto libre como de pago, que proporcionan a los desarrollados una base de ayuda en la implementación de sistemas para robótica móvil. Desgraciadamente, muchas veces este apoyo adicional se reduce a partes concretas de la metodología de trabajo, necesitando una lógica extra que permita integrar las distintas partes, con el consiguiente consumo de recursos en términos de esfuerzo y tiempo.

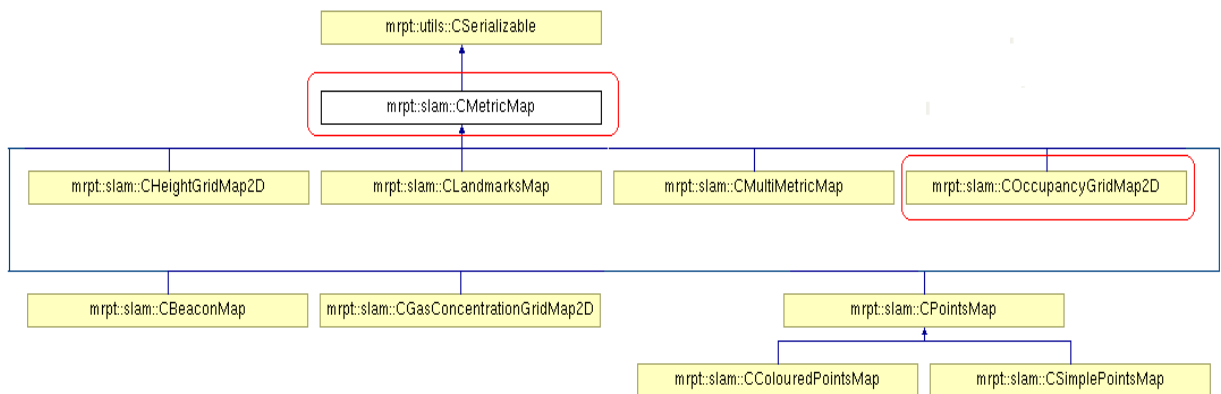
MRPT (*Mobile Robot Programming Toolkit*) nace de la necesidad de aunar distintas herramientas de desarrollo en una única librería extensa, multi-plataforma y de código abierto implementada en lenguaje C ++, buscando en conjunto aumentar la eficacia y la reutilizabilidad de código. Proporciona una base para el diseño de sistemas de navegación, además de disponer de implementaciones en diversos campos, como son el modelado de algoritmos para SLAM, la visión por computador, y la planificación de movimiento para un elemento móvil.

MRPT corresponde a un proyecto desarrollado por el MAPIR (*Machine Perception and Intelligent Robotics Group*), perteneciente al Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga.

No corresponde al ámbito de este proyecto el analizar la estructura de las bibliotecas empleadas, pero sí resulta interesante comentar brevemente algunas de las clases más características cuyo empleo ha sido de crucial importancia en el desarrollo del software. Señalar por último que, a pesar de disponer en ella de mecanismos suficientes para implementar ciertas funcionalidades del proyecto, se ha

preferido que gran parte de las distintas tareas desarrolladas sean controladas a través de herramientas propias, ya sea mediante software creado previamente como mediante la implementación de las acciones que hayan sido necesarias a los largo del desarrollo.

mrpt::slam::CMetricMap



Esta clase virtual sirve como base para la implementación de los distintos tipos de mapas existentes en el entorno de la aplicación. Se puede comprobar la variada cantidad y tipología de mapas que es posible construir a partir de esta clase; el empleo de unos u otros será delimitado por las necesidades del sistema a representar en cuestión, así como de los medios de los que se disponga para las funciones de modelado del medio.

mrpt::slam::COccupancyGridMap2D

Esta clase implementa un mapa del entorno basado en celdas de ocupación, donde cada una de las celdas contendrá un valor probabilístico indicando su grado de ocupación: 0 si la celda está ocupada, 1 si se encuentra libre, 0.5 si existe incertidumbre sobre su contenido. El valor de las celdas se puede modificar de forma independiente.

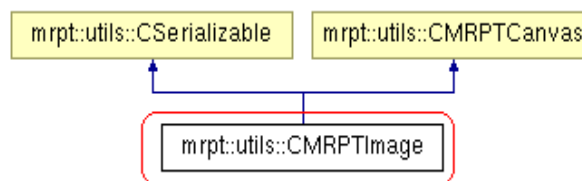
Permite modelar el espacio dividido en celdas rectangulares de igual tamaño. Se puede definir el tamaño del mapa, lo cual permite adaptarlo al entorno tratado en cada momento. Así mismo, dispone de la posibilidad de elegir el tamaño o resolución

que tendrán las celdas. Ambos parámetros, dimensiones del mapa y dimensiones de la celda, mantienen cierta dependencia a la hora de la visualización del mapa.

Los datos correspondientes al espacio que se desea representar son cargados en esta clase desde múltiples formatos. Una de las ventajas ofrecidas por la clase es la posibilidad de cargar la información del entorno desde una imagen ya existente (en formato .jpg, .png, .bmp...) y almacenar los resultados o cambios de la misma manera.

Proporciona algoritmos de búsqueda de caminos, construcción de mapas de Voronoi, filtrado...

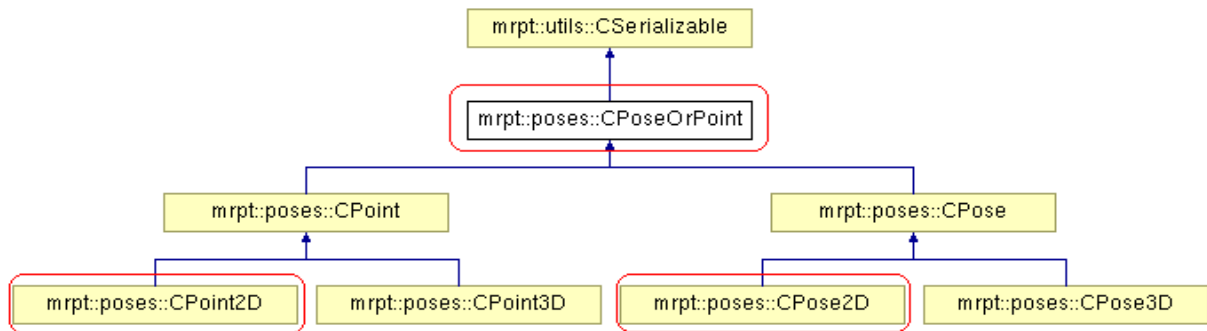
mrpt::utils::CMRPTImage



Esta clase permite almacenar una imagen como un mapa de bits, tanto en escala de grises como en color (RGB – Red Green Blue). Las operaciones de entrada/salida pueden realizarse tanto sobre ficheros binarios como aquellos con extensiones comunes de imagen: .jpg, .png, .bmp...

Permite cambios en la imagen (color, dimensiones, contenido), así como operaciones de filtrado, rectificaciones, correlación de imágenes...

Es empleada para la visualización de mapas, tanto a nivel global como local, ya que permite añadir a los mismos la información gráfica correspondiente al mapeado y la localización, así como al cálculo de rutas.

mrpt::poses::CPoseOrPoint

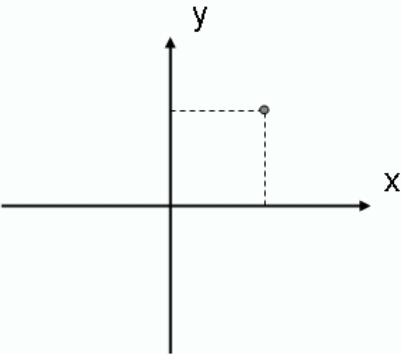
Clase base empleada para representar vectores de posición en 2 ó 3 dimensiones, los cuales a su vez pueden ser puntos del espacio o posiciones dentro del mismo (punto+orientación).

Debido a las características del entorno, bajo la suposición de la no existencia de elevaciones ni irregularidades en el terreno, sólo será necesario emplear parámetros de 2 dimensiones (no se tiene en consideración la altura del medio ni los ángulos asociados a la existencia de dicho eje)

mrpt::poses::CPoint2D

Representa un punto o posición en el espacio cartesiano (x,y), empleándose tanto para este fin como para la representación de las coordenadas gps de un punto (latitud,longitud).

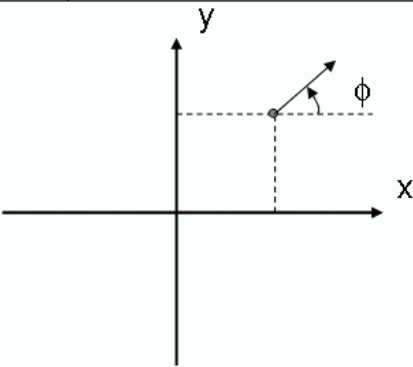
Se implementa en esta clase las posiciones del camino a seguir proporcionadas por el algoritmo de búsqueda global, tanto en el propio mapa total del entorno como en sus correspondencias en los mapas locales obtenidos mediante el proceso de percepción.

<code>poses::CPoint2D</code>	
Homogeneous transformation matrix	Spatial representation
$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

`mrpt::poses::CPose2D`

Representa un punto o posición orientada en el espacio cartesiano (x,y,phi), pudiéndose utilizar también para indicar las coordenadas gps correspondientes (latitud,longitud,phi); siendo phi la orientación o ángulo del robot respecto al Norte geográfico. Dicho valor lo obtenemos mediante el dispositivo gps, y se corresponde al ángulo de viraje (yaw) dado en radianes.

Se implementa en esta clase la posición que sigue el robot, ya que es necesario conocer la dirección a la que se encamina.

<code>poses::CPose2D</code>	
Homogeneous transformation matrix	Spatial representation
$\begin{bmatrix} \cos j & -\sin j & 0 & x \\ \sin j & \cos j & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

6.2 Librerías de soporte

La instalación y puesta en funcionamiento de las librerías que conforman MRPT requiere el empleo de una serie de bibliotecas complementarias, las cuales se indican a continuación:

- OpenCV: muy empleada en diversas aplicaciones, esta librería es una de las más conocidas en el ámbito de la visión artificial.
- Boost: más reciente y menos conocida que la anterior, su empleo en el proyecto se limita en este caso a dar soporte a otras bibliotecas.
- wxWidgets: proporciona una sencilla API para la creación de interfaces de usuario en múltiples plataformas, sin necesidad de cambios en los controles o las utilidades. Entre las ventajas que proporciona, se encuentra la posibilidad de programas multihilo, conversión de imágenes en múltiples formatos, soporte de base de datos, etc.

Para enlazar la aplicación a desarrollar y las distintas bibliotecas, se hará uso de la herramienta conocida como CMake, un sistema multiplataforma que permite la construcción automática de los proyectos en Visual C mediante una serie de ficheros previos de configuración.

6.3 Clases principales empleadas en el desarrollo del sistema

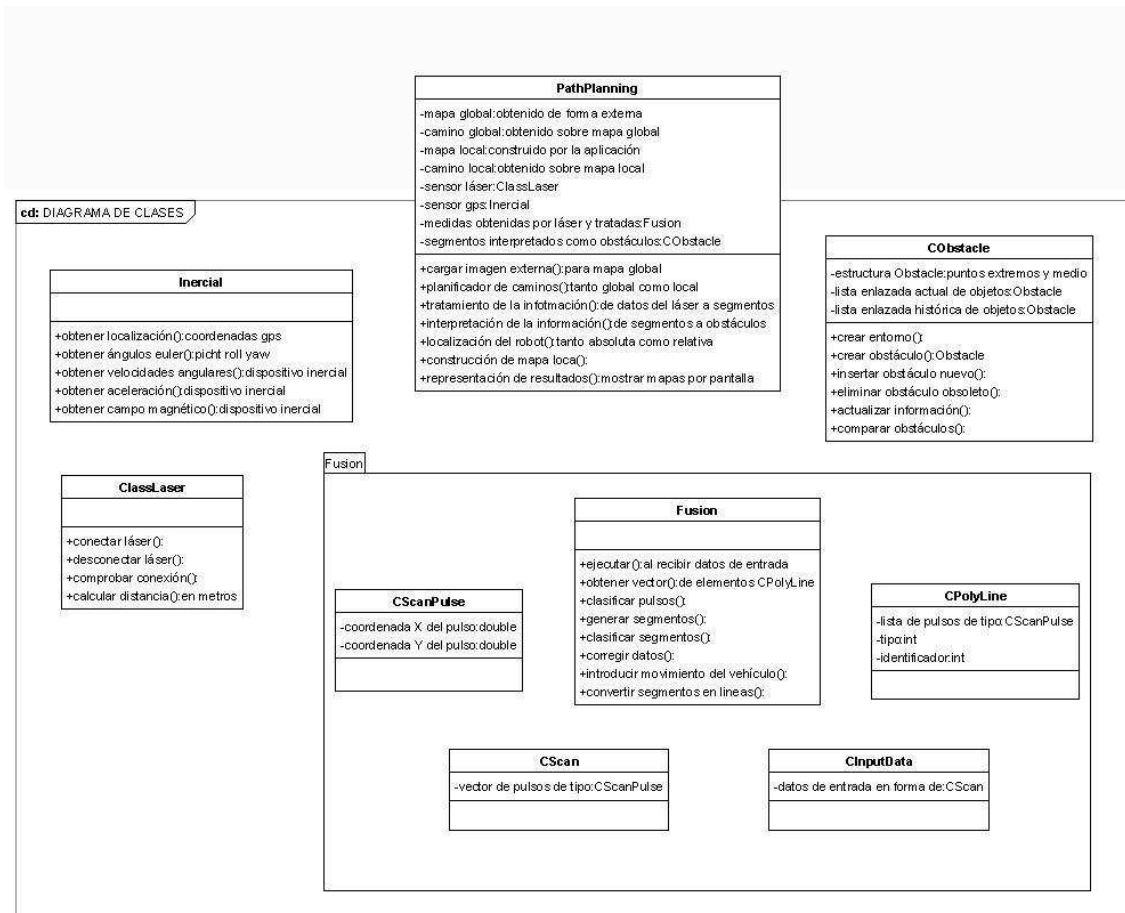


Figura 6-1: clases que componen el sistema

ClassLaser

Esta clase permite la adquisición de la información del medio, ya que representa la interfaz de comunicación con el láser. Permite la configuración del mismo y su conexión al sistema, así como la obtención de los datos de medida (en forma de array de distancias, donde dichas distancias se miden en metros).

Fusion

Esta clase se emplea en el tratamiento de la información proporcionada por el láser, proporcionando métodos para la corrección, clasificación y agrupamiento de los datos. Permite modelar los mismos como estructuras lineales o segmentos.

CObstacle

Esta clase permite interpretar la información tras su tratamiento previo, transformándola en estructuras planas u obstáculos, y ofreciendo un sistema de almacenamiento, comparación y combinación de los mismos.

Los obstáculos así obtenidos serán una representación de las condiciones del entorno vistas por el móvil hasta el momento.

Inercial

Es la clase empleada en el sistema de localización, empleándose como interfaz de comunicación con el dispositivo GPS.

El MTi-g proporciona al programador la biblioteca necesaria para interactuar con el dispositivo, y estos métodos son llamados por la clase Inercial tanto para la configuración del mismo como para la obtención de los datos (de calibración o procedentes de las tramas recibidas vía satélite)

PathPlanning

Esta es la clase principal de la aplicación, conteniendo los mecanismos necesarios para la creación de mapas y la planificación de caminos. Es la encargada de coordinar el resto de funciones, así como de evaluar sus prestaciones. Se encarga así mismo de la configuración adoptada por el robot y los cambios de referencia efectuados a medida que este avanza.

El funcionamiento general de esta clase se representa en la Figura 6-2.

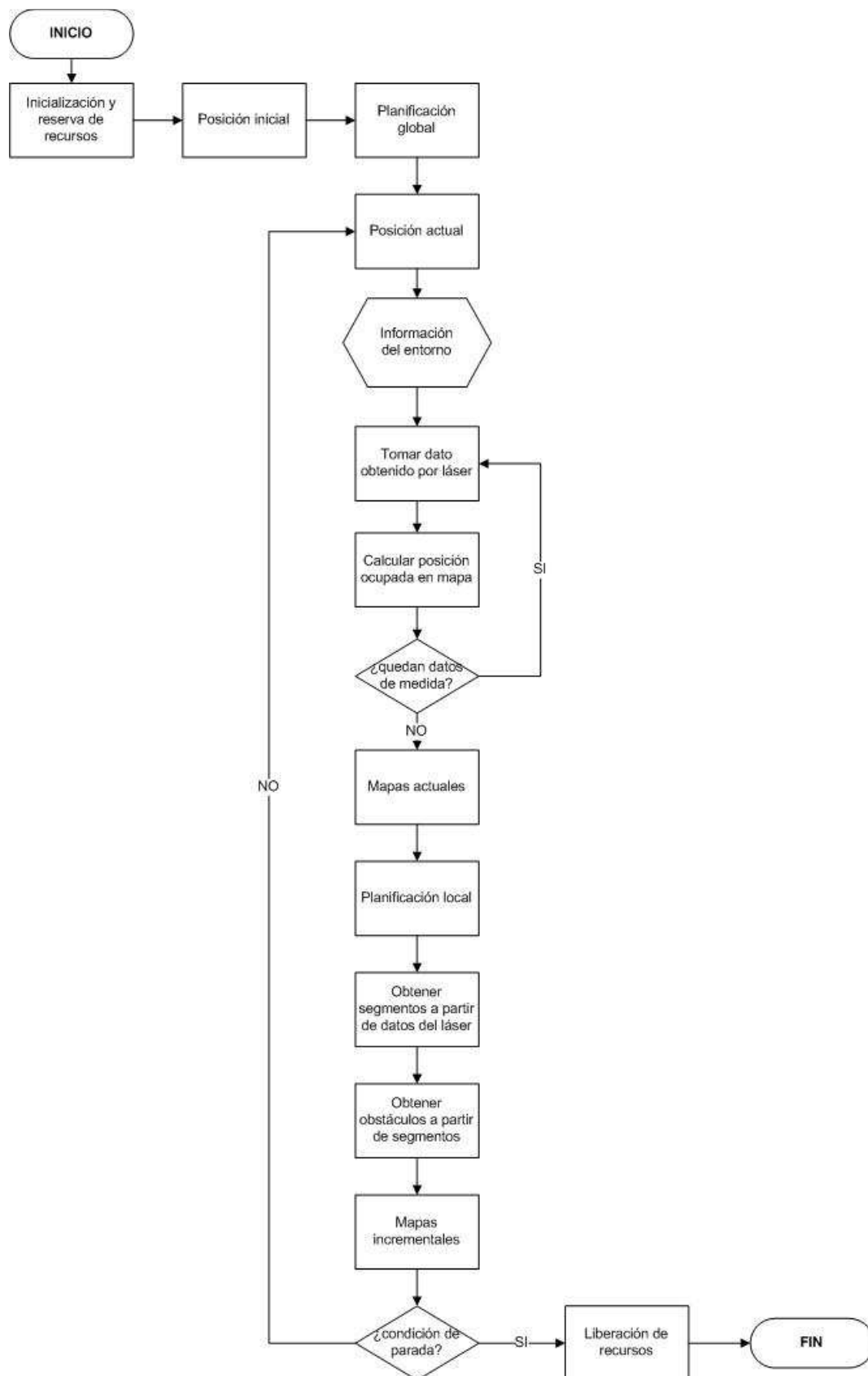
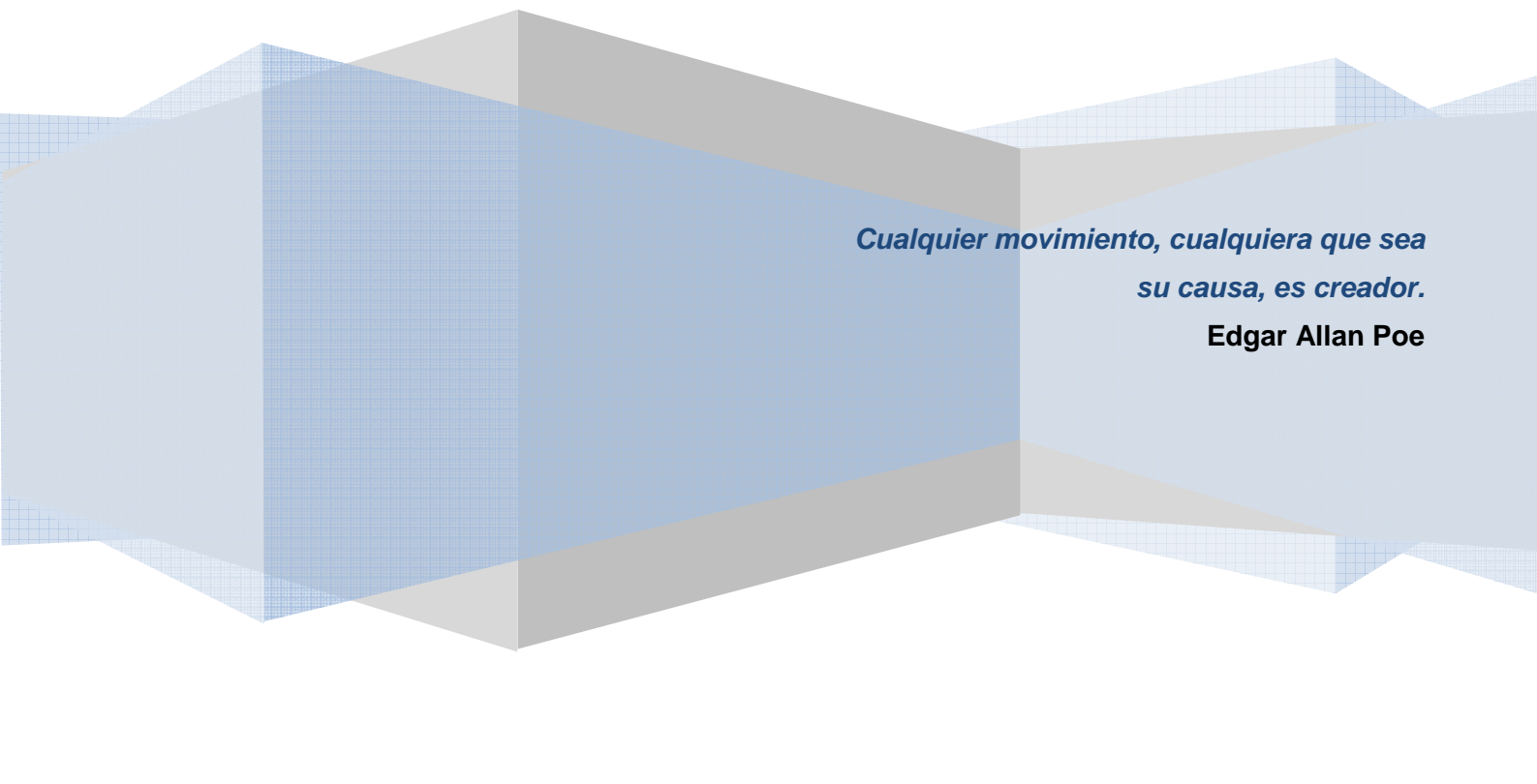


Figura 6-2: esquema general de funcionamiento de la clase PathPlanning

CAPÍTULO 7.

IMPLEMENTACIÓN



*Cualquier movimiento, cualquiera que sea
su causa, es creador.*

Edgar Allan Poe

7. Implementación

7.1 Mapeado mediante celdas de ocupación

7.1.1 Base teórica

El mapa se representa como un campo de variables aleatorias, donde el espacio físico se divide en rejillas de forma uniforme, cada una de las cuales lleva asociada una serie de valores para dichas variables [Thr05]. Los algoritmos de mapeado realizan posteriormente una estimación aproximada de dichas variables aleatorias, calculando la función de densidad de probabilidad en función de los datos obtenidos:

$$p(m|z_{1:t}, x_{1:t})$$

Donde m representa el mapa, $z_{1:t}$ el conjunto de todas las medidas realizadas hasta el instante t , y $x_{1:t}$ es el camino definido a través de la secuencia de todas las posiciones hasta dicho instante. Hay que tener en cuenta que $z_{1:t}$ se encuentra delimitado por los límites de medida del sensor, correspondiendo a aquellas posiciones dentro del ángulo de medida que abarca el láser.

Denominando m_i a la celda de índice i , el mapeado mediante rejillas de ocupación se encargará de dividir el mapa global en múltiples celdas de forma que se pueda expresar como: $m = \{m_i\}$

Cada m_i lleva asociado un valor binario que representa si la celda está ocupada (mediante un “1”) o en su defecto, libre (mediante un “0”). La probabilidad de ocupación de una celda se puede representar pues mediante la fórmula: $p(m_i = 1)$ o simplemente $p(m_i)$

El problema de calcular probabilidades en el mapa completo m radica en las elevadas dimensiones que toma la función de densidad de probabilidad, relacionadas directamente con el número de celdas que constituyen dicho mapa. Así, para un número de rejillas n del orden de miles o decenas de miles de celdas, el número de

mapas que pueden representar dicho escenario es de 2^n , lo que hace inviable el cálculo de probabilidades a posteriori de cada uno de los mapas.

La solución radica en realizar una aproximación al problema global, dividiéndolo en un conjunto de problemas independientes para cada celda m_i , de forma que pueda calcularse la probabilidad a posteriori de cada una de ellas como: $p(m_i|z_{1:t}, x_{1:t})$, lo que reduce cada estimación a un problema binario con estado estático. De esta forma, se puede aproximar el mapa global como:

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t})$$

La ocupación de cada celda i en el instante t se puede modelar como el logaritmo de las probabilidades de la siguiente forma:

$$l_{t,i} = \log \frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})}$$

Donde el valor a priori de ocupación de celdas l_0 toma un valor constante calculado como:

$$l_0 = \log \frac{p(m_i = 1)}{p(m_i = 0)} = \log \frac{p(m_i)}{1 - p(m_i)}$$

La ventaja de esta representación frente a la fórmula tradicional de densidad de probabilidad, es que se pueden evitar las inestabilidades numéricas surgidas en torno a los valores extremos (cero y uno), de forma que podemos expresar las probabilidades a posteriori como:

$$p(m_i|z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + \exp\{l_{t,i}\}}$$

El tratamiento de la información obtenida mediante láser permitirá ajustar los valores de $l_{t,i}$ a los que correspondan en cada momento. Esto lleva a una extensión gradual del mapeado, desde un mapa inicial a la incorporación de los sucesivos

mapas locales obtenidos tras el movimiento (y por tanto, cambios en las zonas medidas) del vehículo.

Además de los datos suministrados por los sensores, es necesario tener en cuenta la información que puede proporcionar el propio robot a la hora de tomar las decisiones de ocupación de las celdas. De esta forma, es necesario considerar que las posiciones adyacentes a aquella x_t donde se encuentra dicho móvil deben ser navegables (celdas libres); así como incluir el volumen del móvil a la hora de generar los mapas, sobre todo en entornos variables y donde existe alta densidad de obstáculos móviles.

7.1.2 Aprendizaje inverso en los modelos de medida

El algoritmo de mapeado mediante rejillas de ocupación emplea en su desarrollo modelos inversos de medida para el cálculo de $p(m|z, x)$ ([Thr05], 2005). Esta probabilidad es denominada “inversa” debido a que realiza el análisis partiendo del efecto y llegando a las causas que lo generan: proporciona información del entorno a partir de las medidas obtenidas en dicho entorno.

La cuestión que se plantea en este punto es cómo es posible obtener modelos inversos partiendo de los modelos de medida tradicionales; el primer paso para ello lleva al desarrollo mediante Bayes de la ecuación de probabilidades:

$$p(m|x, z) = \frac{p(z|x, m)p(m|x)}{p(z|x)} = \eta p(z|x, m)p(m)$$

Se ha asumido que $p(m|x) = p(m)$ o en otras palabras, que la posición del robot no aporta información extra sobre el mapa.

La ecuación anterior se modifica para obtener su equivalente específica para cada celda m_i que conforma el mapa m :

$$p(m_i|x, z) = \eta \sum_{m:m(i)=m_i} p(z|x, m)p(m)$$

Sin embargo, esta suma se hace inviable debido a la elevada dimensión del espacio de mapas, lo que lleva a realizar una aproximación mediante muestreo previo en el modelo de medida y posterior empleo de algoritmos de aprendizaje supervisado, como es el caso de los métodos de regresión o las redes neuronales.

El muestreo consiste en generar de forma aleatoria una serie de tripletas $(x_t^{[k]}, z_t^{[k]}, m_i^{[k]})$ correspondiente a la k -ésima muestra tomada en una misma celda i y en un mismo instante de tiempo t . $x_t^{[k]}$ y $z_t^{[k]}$ corresponden respectivamente, a la posición y a la medida tomadas para esa muestra k -ésima dentro de la celda correspondiente a m_i ; $m_i^{[k]}$ es el valor de ocupación correspondiente a la celda si tenemos en cuenta los dos valores (entradas) anteriores.

7.1.3 Criterio MAP (*Maximun A Posterior Occupancy Grid Mapping*)

Como se ha visto anteriormente, una de las premisas del algoritmo básico de mapeado mediante rejillas de ocupación parte de la hipótesis de independencia entre las celdas que dividen el espacio. Este hecho puede derivar en conflictos entre medidas consecutivas, cuando en una de ellas exista obstáculo para una determinada celda, y en la siguiente no. Es preciso por ello introducir una serie de dependencias en el cono de medida para celdas adyacentes, lo que será detallado a continuación ([Thr05], 2005).

Dicha dependencia es tenida en cuenta mediante la maximización de la probabilidad de ocupación a posteriori en el modelo de mapa completo:

$$m^* = \operatorname{argmax}_m \log p(m|z_{1:t}, x_{1:t})$$

Donde m^* es el mapa estimado.

Aplicando la fórmula de Bayes, se puede desarrollar el modelo anterior de la siguiente manera:

$$\log p(m|z_{1:t}, x_{1:t}) = \log p(z_{1:t} | x_{1:t}, m) + \log p(m) + \text{valor constante}$$

El primer sumando se puede descomponer como suma de las medidas individuales en cada instante de tiempo:

$$\log p(z_{1:t} | x_{1:t}, m) = \sum_t \log p(z_t | x_t, m)$$

El segundo término, correspondiente a la probabilidad a priori de cualquier mapa m , se puede desarrollar como:

$$\begin{aligned} p(m) &= \prod_i p(\mathbf{m})^{m_i} (1 - p(\mathbf{m}))^{1-m_i} = (1 - p(\mathbf{m}))^N \prod_i p(\mathbf{m})^{m_i} (1 - p(\mathbf{m}))^{-m_i} \\ &= \eta \prod_i p(\mathbf{m})^{m_i} (1 - p(\mathbf{m}))^{-m_i} \end{aligned}$$

$p(\mathbf{m})$ es la probabilidad a priori de ocupación de las celdas (cuyo valor inicial puede ser, por ejemplo, 0.5), N el número total de celdas en el mapa, y η es una constante que equivale a la expresión $(1 - p(\mathbf{m}))^N$, cuyo valor es constante una vez fijada la probabilidad a priori.

Aplicando logaritmo a la fórmula anterior, se llega a:

$$\begin{aligned} \log p(m) &= \sum_i m_i \log p(\mathbf{m}) - m_i \log(1 - p(\mathbf{m})) + \text{constante} = \\ &= \sum_i m_i \log \frac{p(\mathbf{m})}{1 - p(\mathbf{m})} = \sum_i m_i l_0 + \text{constante} \end{aligned}$$

Aplicando los desarrollos anteriores a la fórmula inicial de maximización de las probabilidades a posteriori, se concluye que:

$$m^* = \underset{m}{\operatorname{argmax}} \sum_t \log p(z_t | x_t, m) + l_0 \sum_i m_i$$

7.2 Integración de componentes al proyecto

En los capítulos 4 (*Percepción del entorno físico*) y 5 (*Localización en el entorno físico*) se explicaron los pasos previos para obtener la información del entorno y de la situación del robot en el mismo, necesarios para la construcción de los mapas y el cálculo de caminos. Las acciones realizadas se resumen en la Figura 7-1, donde se representan los casos de uso o funcionalidades presentes en el sistema.

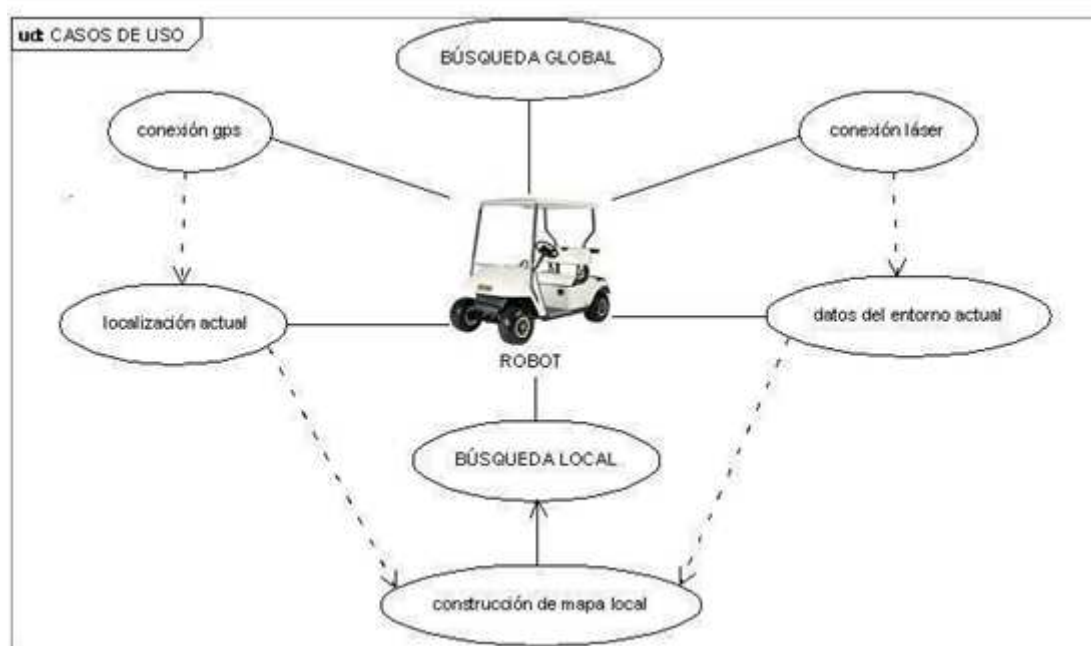


Figura 7-1: casos de uso del vehículo autónomo

A continuación se tratarán los aspectos relacionados con el mapeado y los algoritmos de búsqueda, tanto de forma global como local. Cada uno de estos tipos de planificación conlleva necesidades y acciones distintas, cuyo esquema general se muestra en la Figura 7-2.

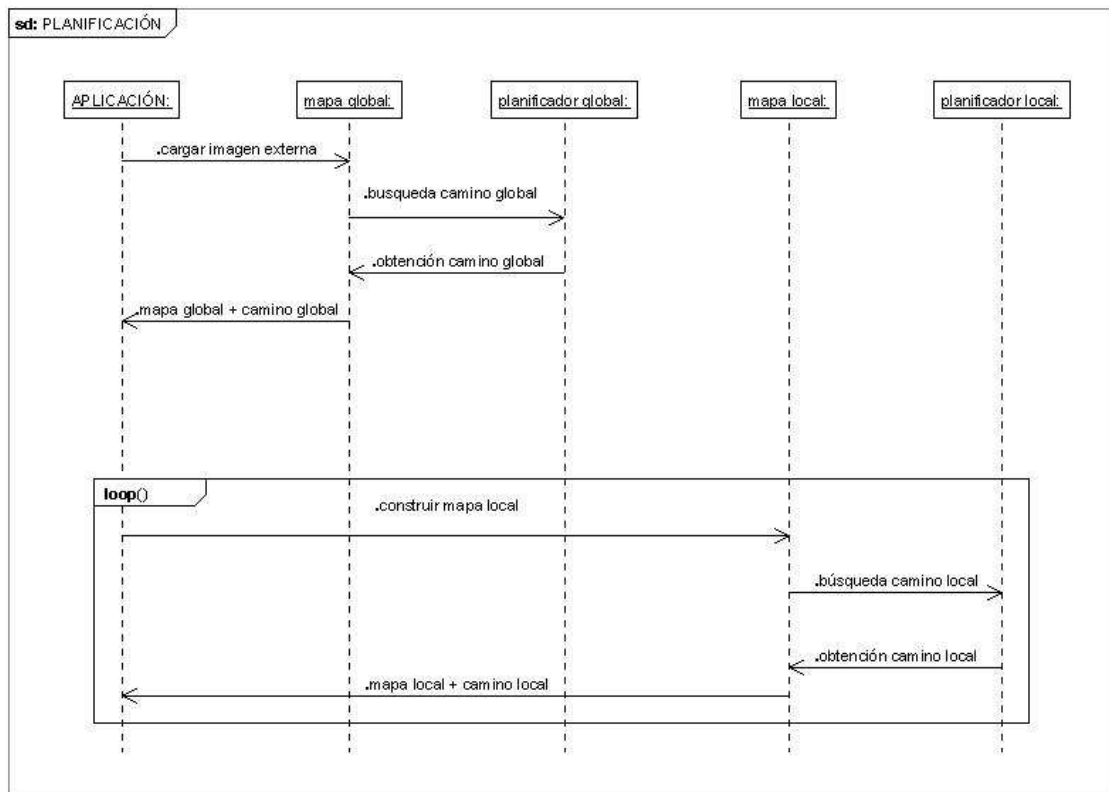


Figura 7-2: secuencia básica de planificación

7.3 Algoritmos de búsqueda global

7.3.1 Mapeado global del entorno mediante celdas de ocupación

El primer asunto a considerar es la obtención de un mapa global del entorno que, a grandes rasgos, sirva para direccionar el movimiento del vehículo. En dicho mapa no es tan importante la precisión de las posiciones a alcanzar, sino la posibilidad de que el móvil alcance un entorno cercano al punto deseado con un margen de error que se pueda considerar razonable para que la desviación no influya decisivamente en la consecución del camino total. Es decir, se pretende que el móvil pase por una serie de puntos marcados previamente de forma más o menos aproximada.

La obtención del mapa global es un aspecto que sobrepasa los objetivos de este proyecto, y cuya construcción sería una propuesta interesante en implementaciones futuras. De los distintos mapas (métricos, topológicos y semánticos) que se describieron en el capítulo 2.3, se ha optado por el empleo de mapas métricos

al ser estos los más adecuados en el entorno de trabajo. El motivo de esta elección viene determinado principalmente por las características del entorno (que en este caso es conocido a priori): de forma simplificada, sólo se hace necesario a grandes rasgos un esquema del mismo en escala de grises, donde el camino libre estaría en blanco y los grandes obstáculos fijos (ya sean paredes, vallas, portales, y todos aquellos elementos estáticos fácilmente identificables del entorno) estarían en negro.

Aunque estas características harían más conveniente el empleo de un mapa métrico geométrico, el tratamiento posterior del algoritmo de búsqueda exige la implementación de mapas basados en rejillas de ocupación, como se explicará en el siguiente apartado.

El sistema de mapeado global sólo precisa como entrada una imagen corriente, fácilmente obtenible mediante Google Maps u otras aplicaciones similares, que posteriormente ha de ser tratada por herramientas externas a la aplicación para distinguir las zonas libres y ocupadas (Figura 7-3); en la realización del proyecto, se escogió una sección trasera de la Universidad, precisamente aquella donde fueron realizadas las pruebas, y se obtuvo la imagen mediante Google Maps. Dicha imagen fue tratada mediante el Paint para que los edificios, vallas y zonas de no tránsito quedasen en negro; el resto, se pintó en blanco (Figura 7-4).

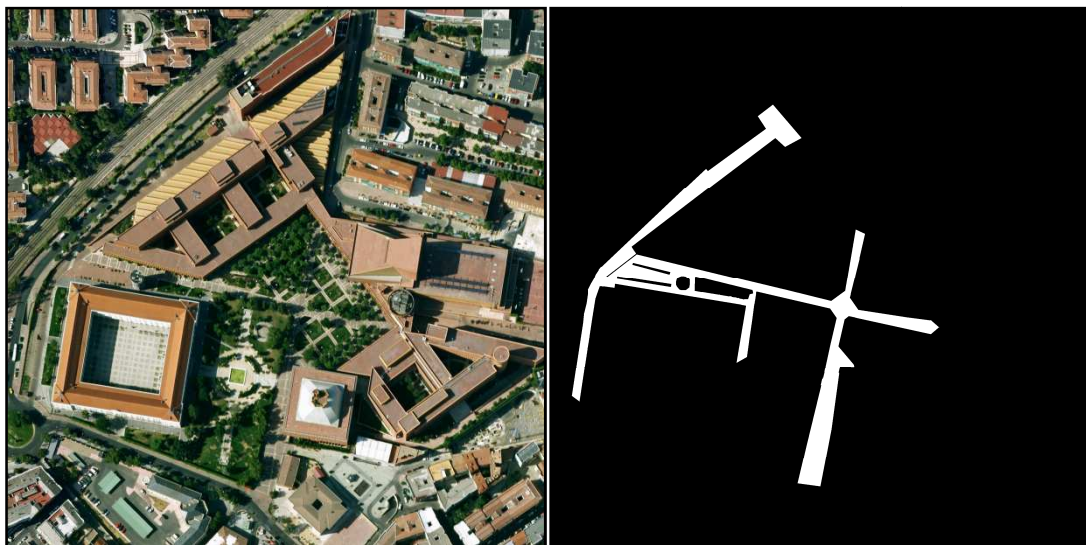


Figura 7-3: imagen completa del campus

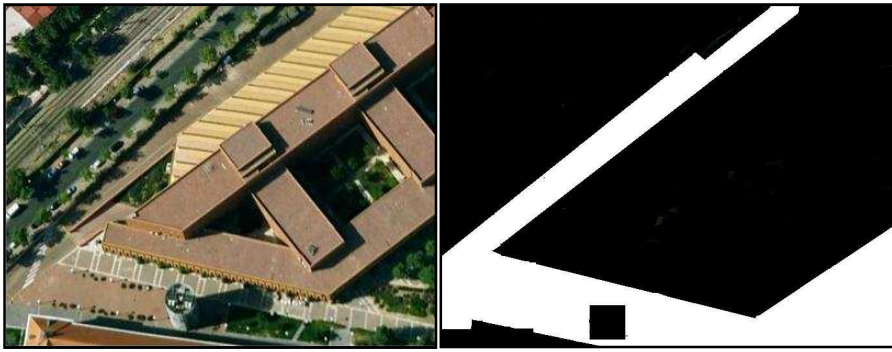


Figura 7-4: imagen de la zona de pruebas

La imagen es cargada por el sistema y convertida en un mapa de probabilidades, donde cada rejilla, blanca o negra, tiene un valor que determina si se trata de una zona de tránsito o de no tránsito.

El mapa obtenido divide el espacio en celdas cuadradas de igual tamaño, y la navegación a través de dicho mapa puede realizarse de diversas formas. Una de ellas consiste en considerar el mapa como una matriz, donde cada celda ocupa una posición consecutiva, y recorrerla por filas. Sin embargo, este sistema no resulta lo más asequible para el usuario, además de no proporcionar información espacial sobre los puntos de acceso. Es por ello que se opta por una representación en coordenadas cartesianas, fijando los valores límites a los reales del entorno, y tomando las medidas en metros. Se trata pues de un sistema de referencia relativo, centrado en el punto medio del espacio total por donde se realiza la circulación del robot. De esta forma, el método de búsqueda de caminos devuelve los puntos por los que ha de pasar el móvil en forma de coordenadas cartesianas, lo que facilita su visualización en el mapa.

De forma posterior, para el tratamiento de dichos puntos por parte de la lógica del vehículo, los puntos son transformados a sus correspondientes coordenadas GPS. Las distintas posiciones del camino, transformadas ahora en duplas (latitud, longitud), proporcionan información absoluta sobre el recorrido a seguir por el móvil.

Es importante señalar que en la transformación a coordenadas GPS, los puntos del camino no incorporan inicialmente información sobre la orientación de los mismos, al tratarse de valores absolutos e independientes entre sí.

7.3.2 Planificación global

Se ha optado por emplear un algoritmo de búsqueda de caminos proporcionado por las bibliotecas, al satisfacer éste los requisitos de búsqueda por gradiente y adaptarse de forma correcta al sistema de mapeado global. En dicho esquema, el robot debe ser modelado como un elemento circular, cuyo radio es un parámetro configurable R_G . Dicho valor no sólo sirve para el cálculo del área que abarca el vehículo, sino que también es empleado como margen de seguridad ante la proximidad de obstáculos, ampliando las zonas ocupadas por los mismos para evitar colisiones.

En la Figura 7-5 se muestra un ejemplo de los resultados obtenidos para distintos valores de R_G . Al disminuir su valor, el modelo de vehículo se aproxima al de un elemento puntual, lo que puede provocar conflictos en zonas estrechas del recorrido, ya que no se tendría en cuenta las dimensiones reales de los elementos implicados (el vehículo intentará llegar a posiciones accesibles en teoría, pero inaccesibles para él). Si se prueba a aumentar gradualmente el valor del radio, llegará un momento en el que el algoritmo de búsqueda no será capaz de encontrar una solución que satisfaga los requerimientos impuestos (hay que recordar que no sólo aumenta el área del vehículo, sino también el de los obstáculos, por lo que se reduce la zona de tránsito).

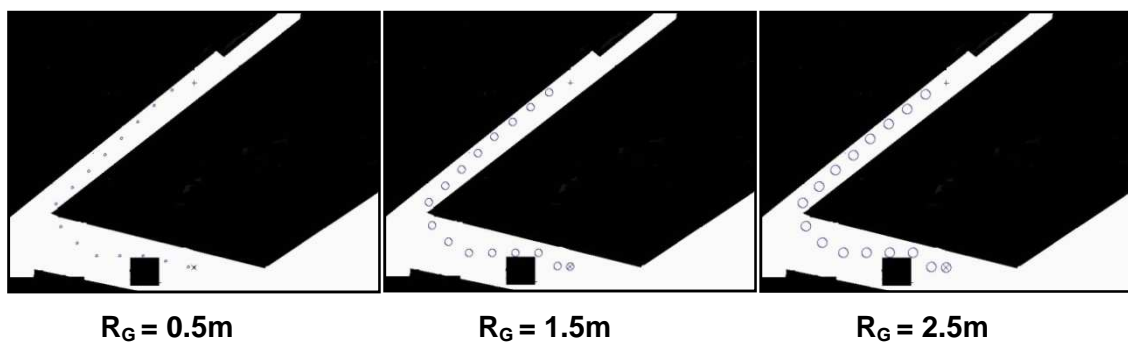


Figura 7-5: camino global para distintos valores de R_G

Esta situación hace necesario un ajuste correcto del valor del radio, que dependerá de las dimensiones del dispositivo móvil y del entorno específico donde tenga lugar la planificación. En el presente desarrollo, el medio presenta amplias

zonas libres y obstáculos bien definidos, con poca presencia de regiones de paso estrecho, lo que permite aumentar R_G hasta un máximo de aproximadamente 3m. El valor elegido finalmente para el radio será de 2.5m, ya que satisface los requisitos tanto del entorno como de las dimensiones del vehículo.

Inicialmente, sólo son conocidos los puntos origen y destino; el resto de puntos del camino serán proporcionados por el algoritmo de planificación. La distancia entre ellos es un parámetro regulado por el usuario, y de este dependerá el número de marcas de camino obtenidas. Un número elevado de ellas proporciona mayor precisión a la hora de recorrer la ruta marcada, pero en la práctica añade una dificultad extra, al tener el móvil que recorrer todos y cada uno de los puntos definidos, cuando la mayoría de ellos ofrece información escasa o redundante sobre cómo es el camino; incluso puede llevar a conflictos si el móvil no puede llegar a posiciones que son de más difícil acceso y no relevantes para llegar a la meta.

Por otro lado, un número limitado de marcas en el camino harían que el móvil perdiese la ruta y dificultaría la aplicación posterior de algoritmos de búsqueda local, ya que las distancias entre puntos de acceso obligatorio podrían sobrepasar los límites de alcance de los mapas locales. Es por ello que se hace necesario encontrar un consenso entre ambas necesidades (suficiente cantidad de marcas y distancia entre ellas)

En la Figura 7-6 se muestran el número de marcas obtenidas para distintos valores de separación o pasos entre posiciones (dados en metros). Como se puede comprobar, la relación entre ambos factores no es lineal (Figura 7-7), produciéndose una mayor concentración de marcas a medida que el paso disminuye, y convergiendo a una cantidad insuficiente de ellas cuando se toman separaciones mayores. También se han representado los tiempos transcurridos en cada búsqueda (Tabla 7-1), cuyos valores no muestran gran diferencia entre sí, por lo que no resultan determinantes en la elección del paso.

valor del paso (metros)	longitud del camino (nº posiciones)	tiempo ejecución (ms)
0.5	258	338.243
1	136	305.636
1.5	99	325.149
2	76	331.597
2.5	63	335.625
3	51	336.602
4	40	307.359
5	32	306.167
6	27	307.89
8	20	308.269
10	16	308.682
12	14	304.291
14	12	312.891
16	10	314.877
18	9	307.8
20	8	321.602
24	7	304.18
28	6	339.147
32	5	305.04
36	4	331.984
40	4	321.613
44	4	331.413
60	3	308.355

Tabla 7-1: longitud del camino encontrado y duración de la búsqueda para distintas separaciones entre posiciones consecutivas

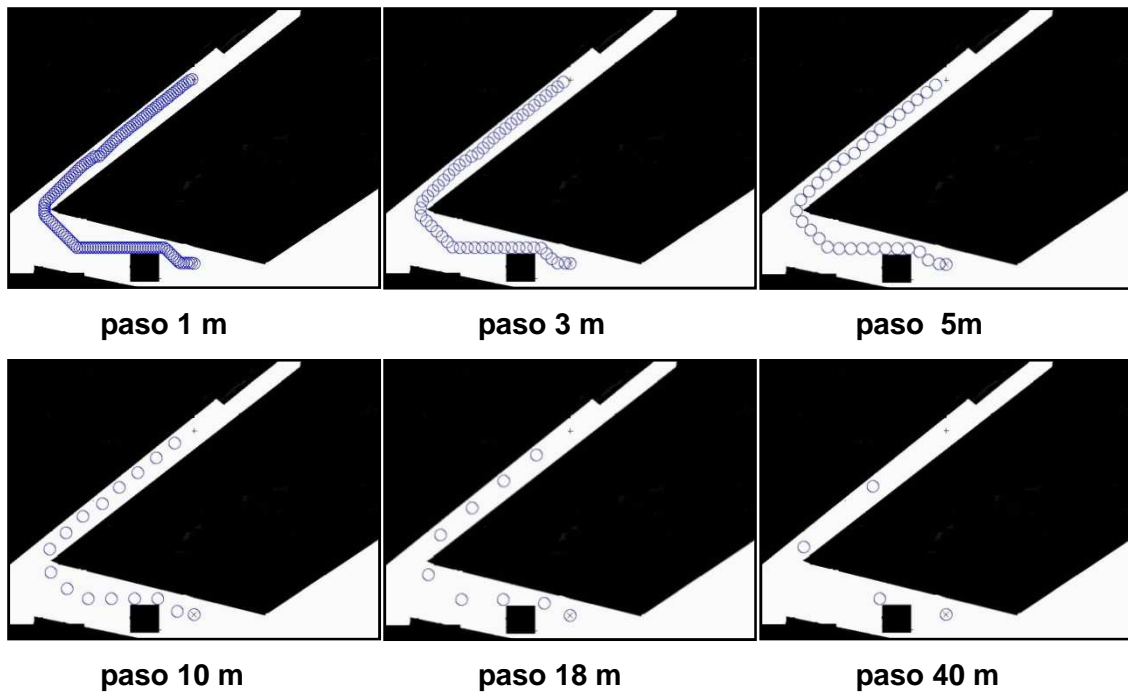


Figura 7-6: caminos globales para distinta separación entre posiciones

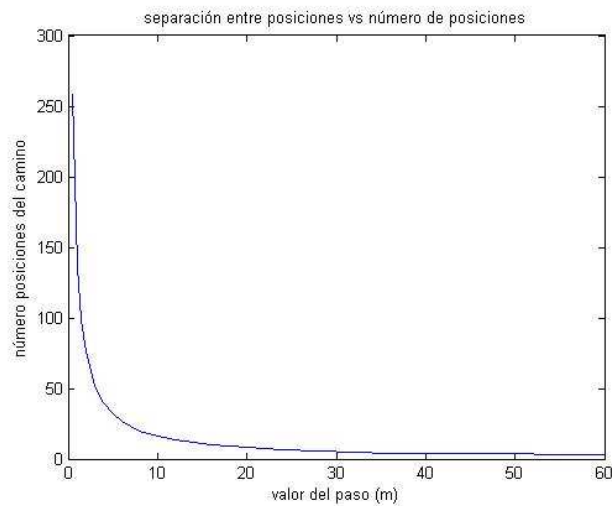


Figura 7-7: separación de posiciones vs número de posiciones

Debido al valor del radio del robot, y a las dimensiones y simetría del medio, las opciones más adecuadas parecen ser las correspondientes a pasos entre 8 y 12 metros. El valor seleccionado pertenecerá por tanto a este margen, eligiéndose finalmente una separación entre posiciones adyacentes de 10 metros.

7.4 Algoritmos de búsqueda local

7.4.1 Mapeado local del entorno mediante celdas de ocupación

El modelado local, a diferencia del global, requiere una serie de pasos previos para obtener la información necesaria del entorno, implicando las operaciones de percepción (apartado 4) y localización (apartado 5), y los dispositivos láser (apartado 4.2.1) y GPS (apartado 5.2) asociados a las mismas; el esquema general se muestra en la Figura 7-8.

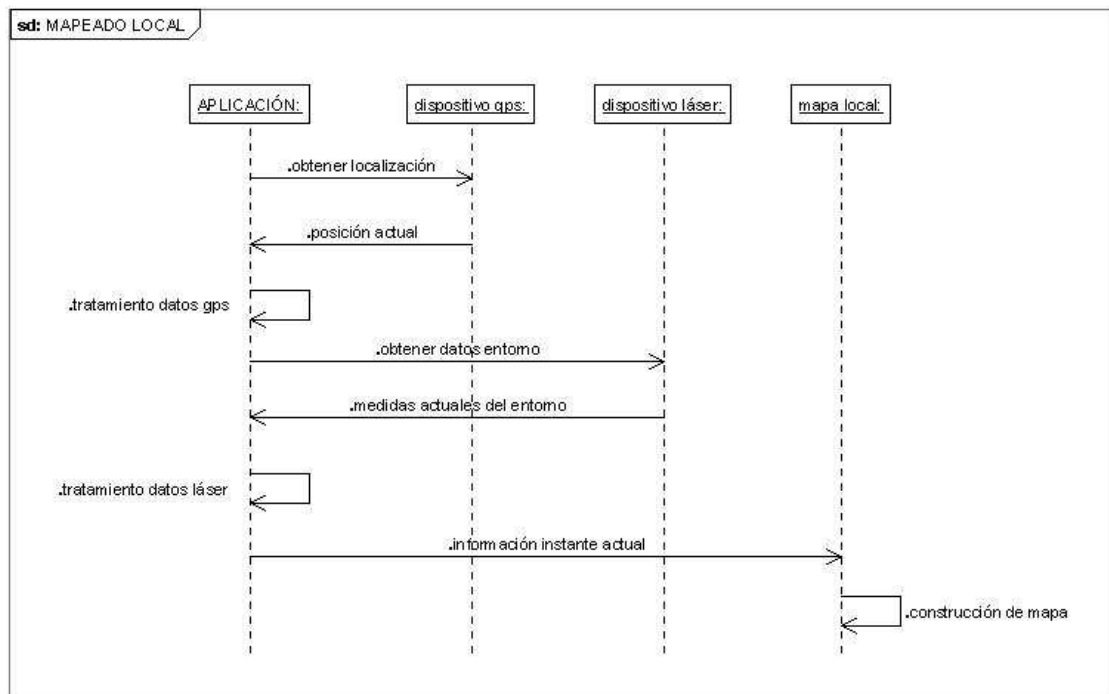


Figura 7-8: secuencia básica del mapeado local

Los mapas locales consisten en planos bidimensionales referenciados por dos ejes perpendiculares, que para facilitar su comprensión se consideran como eje X y eje Y. Al igual que en el mapa global, se trata de mapas de rejillas de ocupación, formados por celdas uniformes. La ocupación de las celdas puede tratarse desde dos vertientes: modelándola como una función de probabilidad (con valores en un intervalo entre 0 y 1), obteniéndose imágenes del entorno en escala de grises; o como un mapa binario (valores 0 y 1 exclusivamente) cuando las características del medio permitan aplicar esta simplificación, dando lugar a imágenes básicas en blanco y negro. En cualquier caso, los mapas se pueden modelar mediante una serie de parámetros,

siendo algunos de los más importantes las distancias máximas y mínimas en los ejes X e Y y la resolución o tamaño de las celdas.

El principal problema de la resolución radica en la visualización posterior del mapa: parecería conveniente, dadas las dimensiones del coche, tomar una resolución mínima correspondiente al diámetro del mismo (modelando el vehículo como una superficie esférica). Sin embargo, con dicho valor obteníamos un mapa demasiado pequeño para poder distinguir correctamente la situación del coche y su entorno, por lo que han sido necesarias distintas pruebas hasta obtener un equilibrio entre el tamaño del coche y la visualización del mapa. La resolución se determina finalmente en 0.25 metros, lo que hacen necesarias 16 celdas para obtener una superficie de 1 metro cuadrado.

Tipo de mapas locales

En cuanto a cómo se representa la información, se ha optado por mostrar 5 tipos de mapas, los cuales se explican a continuación:

I. Mapa del entorno local del vehículo, con un alcance de 80 metro, correspondiéndose con la distancia máxima de medición del láser. Corresponde a lo que realmente ve el robot en el instante de tiempo actual (Figura 7-9).

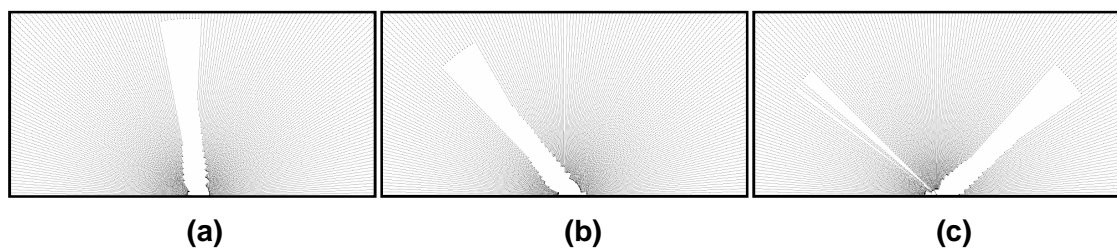


Figura 7-9: mapas locales con el láser (a) apuntando al frente, (b) girado a la derecha, (c) girado a la izquierda

II. Mapa del entorno local próximo al vehículo, con un alcance de 16 metros. Idéntico al anterior, pero reduciéndose el área para mejorar la visualización (Figura 7-10). Se hizo necesario implementar este mapa de forma independiente, ya que la herramienta MRPT no ofrecía posibilidad de realizar un zoom del mapa anterior.

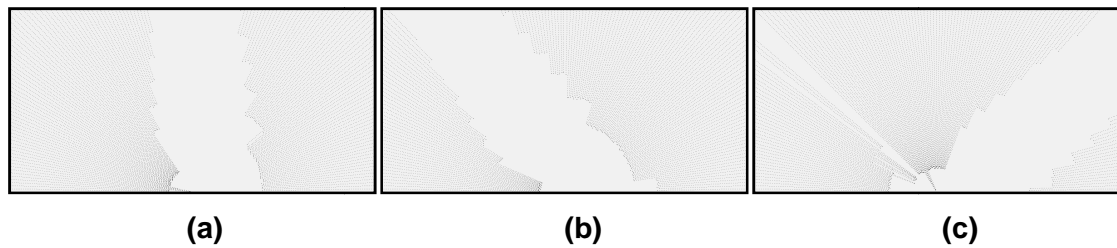


Figura 7-10: mapas de proximidad con el láser (a) apuntando al frente, (b) girado a la derecha, (c) girado a la izquierda

III. Mapa de evolución del movimiento, que muestra el entorno local del vehículo en el instante actual en relación al medio por el que se desplaza (Figura 7-11). Al igual que los mapas anteriores, sólo contiene información del presente, pero es la base para la planificación local y para la obtención de mapas más elaborados, como el mapa incremental del camino y el mapa de obstáculos.

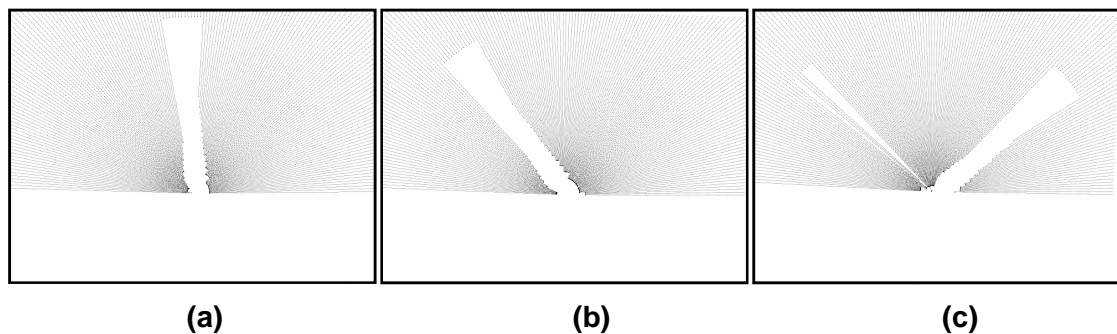


Figura 7-11: mapas de movimiento con el láser (a) apuntando al frente, (b) girado a la derecha, (c) girado a la izquierda

IV. Mapa incremental, con toda la información que el vehículo ha visto hasta el momento. Su estructura e implementación difieren por completo de los anteriores, mostrando el camino real seguido por el móvil (Figura 7-12). Es un mapa que resulta especialmente útil e interesante una vez concluida la navegación, pudiendo emplearse de forma posterior para realizar el mismo recorrido (o alguno similar).

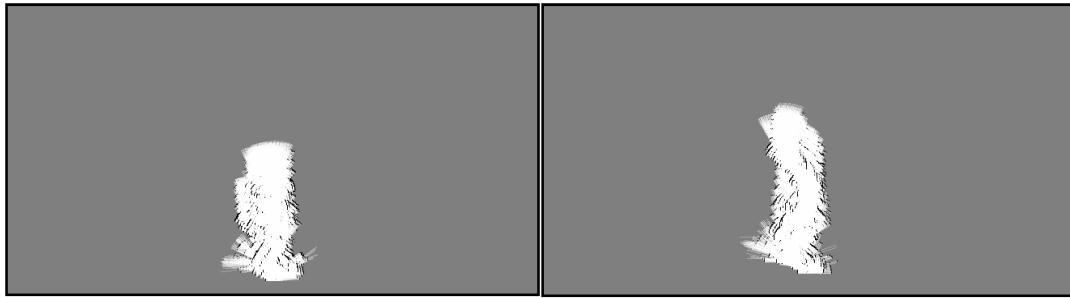


Figura 7-12: mapas incrementales

V. Mapa de obstáculos, donde la información de las mediciones realizadas hasta el instante actual se modela para formar entidades con significado propio, generando un conjunto de elementos que representan los obstáculos del entorno (Figura 7-13). Como en el caso anterior, su verdadero potencial se hace patente al finalizar el recorrido.

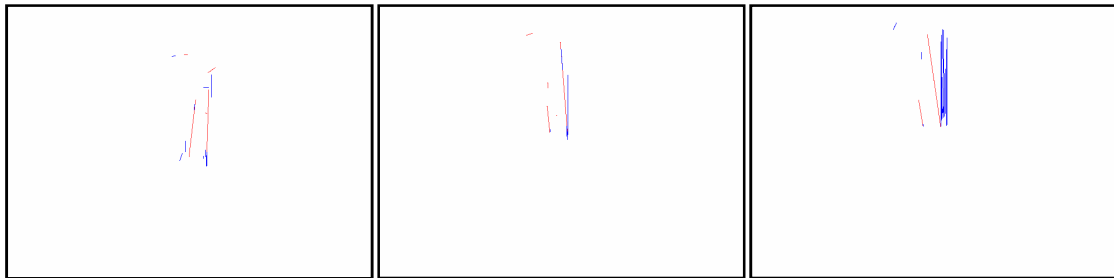


Figura 7-13: mapas de obstáculos

El motivo de tal variedad se debe a las distintas necesidades surgidas a lo largo del proyecto: al tratarse éste de una fase de desarrollo previa a la implementación final, se han decidido estudiar varios casos para elegir entre ellos aquel que resulte más adecuado.

Desde un punto de vista estético se ha querido, por un lado, representar el entorno como si el usuario pudiese ver a través de los *ojos* del robot; por otro lado, se pretende la visualización completa del espacio recorrido (una recreación del mapa global a partir de los datos locales obtenidos en tiempo de ejecución); desde un punto de vista práctico, se pretende resolver el problema de la navegación del vehículo. Al

tratarse de mapas de distinta naturaleza y utilización, han sido necesarias distintas técnicas de modelado, como se explica a continuación. Para una mejor comprensión de los mismos, se separarán los mapas que sólo reflejan los datos tomados en un instante determinado (tipos I, II y III) de los que almacenan la información histórica del recorrido o mapas incrementales (tipos IV y V).

Mapas del instante actual

Los mapas del entorno actual del móvil (I y II) no tienen otra función que la de mostrar lo que realmente se está viendo desde esa posición; mientras que el mapa del robot a medida que evoluciona su posición (III) es el que verdaderamente recoge la problemática de resolver el camino, y es donde, en la práctica, se aplicará el algoritmo de búsqueda local.

Para la construcción de estos tres tipos de mapas, se parte de la suposición de entorno libre de obstáculos, y a partir de ahí se produce la clasificación de los puntos obtenidos mediante muestreo del láser, cuyas celdas correspondientes toman un valor máximo de ocupación (probabilidad '1') y se representan en color negro. Posteriormente, se tratan los puntos correspondientes a cada haz, a partir del obstáculo, con un valor de ocupación de 0.5 y un muestreo de 0.5 metros.

En un primer momento se intentó no muestrear el espacio de celdas, sino tratar individualmente todas y cada una de ellas, de forma que quedase una imagen espacial homogénea. Sin embargo, debido al tamaño reducido de las mismas y a las características del mapa (siendo necesario recorrerlo como matriz), el tiempo de cómputo no resultaba factible en la práctica, además de tratarse de regiones de las cuales no se posee información veraz. Por lo tanto, se ha procurado reducir su tratamiento de la forma más rápida y factible posible (buscando un acuerdo entre velocidad de cómputo y reconocimiento del entorno).

El valor de ocupación es fijado a 0.5 para indicar la incertidumbre en la asignación de probabilidad, ante la incapacidad de distinguir qué ocurre con las celdas más allá de las percibidas como ocupadas; dichas celdas pueden estar ocupadas o no dependiendo del área ocupada por el obstáculo, del cual no poseemos información.

Como se explicó en el capítulo 4.2, el proceso de percepción puede ser desarrollado en tres fases: adquisición de datos, tratamiento e interpretación. En el proceso de mapeado correspondiente a los mapas de tipo I, II y III, el único paso obligatorio para modelar el entorno consiste en la adquisición de las características del mismo, ya que aporta la información mínima necesaria para la construcción del mapa (en el caso del mapa III, además se añade información sobre la posición del vehículo).

Mapas con datos históricos

Donde realmente resulta interesante aplicar las técnicas de tratamiento e interpretación de la información es en el caso de mapas que integran todo el conocimiento obtenido hasta el momento, como es el caso del mapa incremental (IV) y el mapa de obstáculos (V). Por este motivo, es aquí donde se realiza la actualización de la información acumulada.

El mapa incremental (IV) es obtenido mediante técnicas probabilísticas implementadas por la biblioteca MRPT. Partiendo de un entorno desconocido a priori (modelando la incertidumbre a un valor de 0.5), la probabilidad de ocupación de las celdas va variando en cada nueva iteración en función de la información actual y de la obtenida anteriormente, aplicando las técnicas vistas en el capítulo 7.1.

Su principal ventaja radica en el hecho de realizar el tratamiento tanto de las zonas de colisión como de las zonas libres. En este caso particular, los datos del láser, conocidos con el nombre de *observaciones*, delimitan no sólo los obstáculos, sino también las regiones de tránsito.

La Figura 7-14 muestra una secuencia de mapas tomados en instantes consecutivos de tiempo. En este caso, el robot no ha experimentado traslación, sólo rotación a derecha e izquierda; se puede comprobar cómo, a medida que aumentan las medidas en cada uno de los casos (a, b, c), la imagen se va volviendo más definida, al concretarse las zonas de libre tránsito.

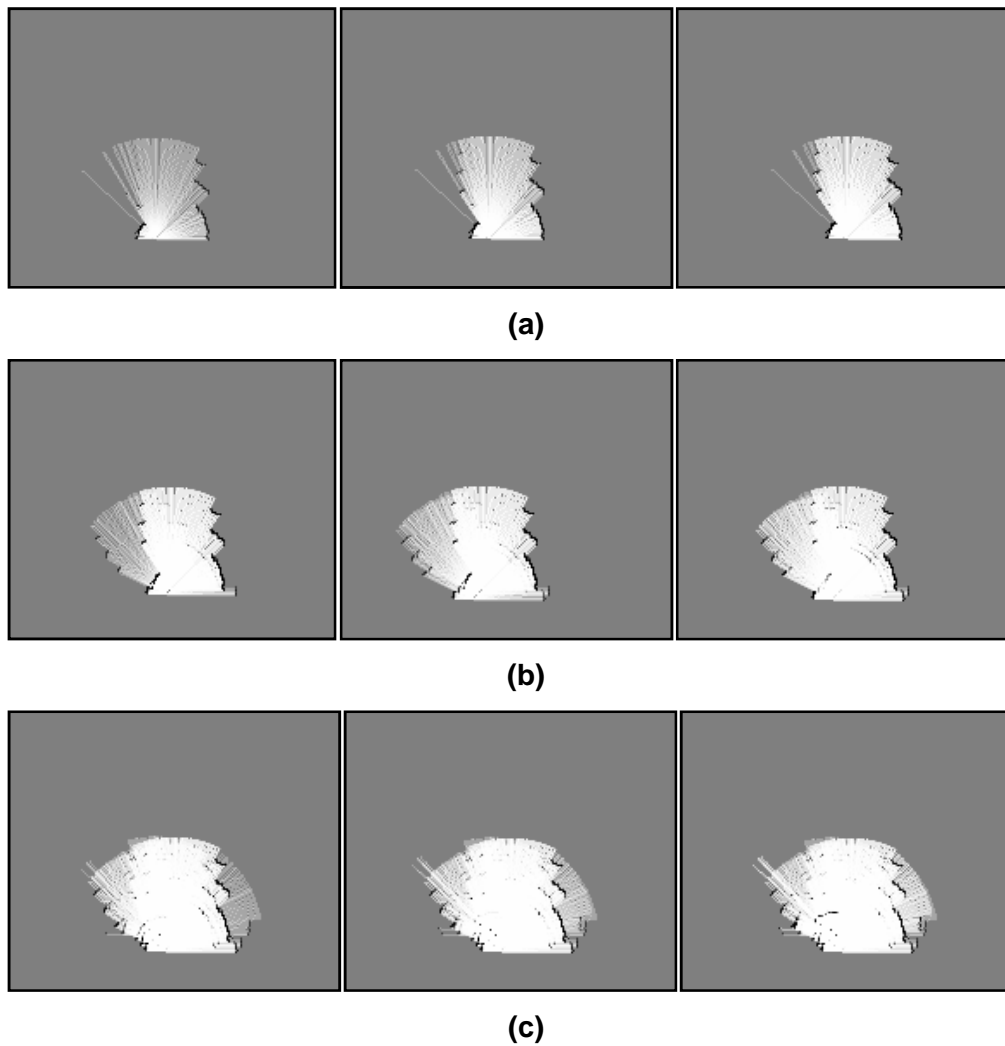


Figura 7-14: mapas incrementales con el láser (a) apuntando al frente, (b) girado a la derecha, (c) girado a la izquierda

Sin embargo, y a pesar de sus prestaciones, no se pudo hacer uso en exclusiva de este modelo para la tarea de planificación local, debido a dos factores fundamentales: los datos erróneos de medida en el láser y, sobre todo, los problemas derivados de la localización del vehículo. Este tipo de mapeado presenta gran sensibilidad a imprecisiones en la localización, sobre todo ante la presencia de ‘saltos’ bruscos en la posición del vehículo (debido a la deriva de los datos proporcionados por el GPS). La Figura 7-15 muestra un ejemplo de este tipo de situaciones, tanto con el robot parado como en movimiento.

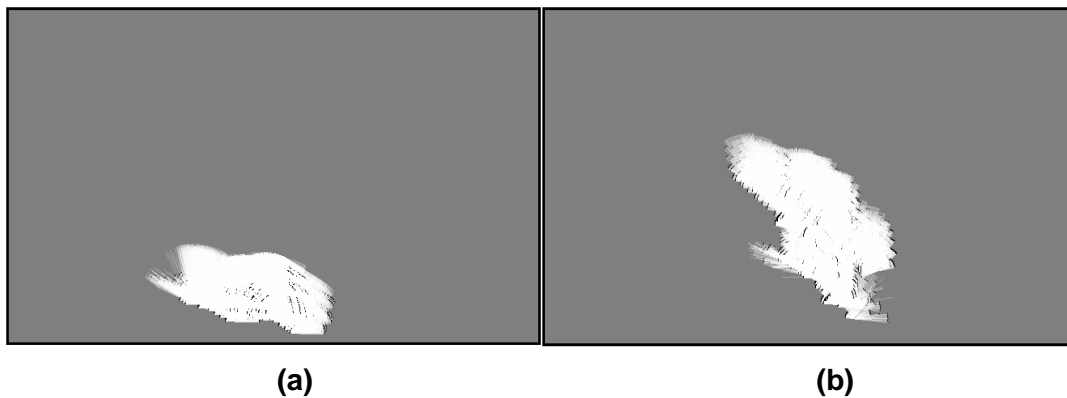


Figura 7-15: deriva de la posición, con el robot parado (a) y moviéndose en línea recta (b)

Estos problemas intentarán ser solventados por el mapa de obstáculos (V), que también se considera como incremental al modelar la información anterior al momento actual. Como mecanismo de robustez frente a errores en los sensores, se realiza la interpretación del entorno de forma selectiva, buscando pautas comunes y desechando la información no regular o desactualizada. Debido a su complejidad, este modelo se explicará con más detenimiento en el capítulo 7.5.

La Figura 7-16 representa una secuencia de mapas de obstáculos, tomada en instantes consecutivos de tiempo, sin realizarse traslación del vehículo (sólo rotación). En color rojo, se representan los obstáculos actuales (obtenidos en cada iteración); en azul, los obstáculos anteriormente obtenidos (almacenados en memoria). En la imagen (a) se puede comprobar cómo, ante una situación estable, los obstáculos son refrescados continuamente, de forma que existe correspondencia entre cada uno de los nuevos y los antiguos. Al producirse un cambio en las condiciones del robot (rotación a derecha - b - y a izquierda - c -), durante un intervalo de tiempo aparecerán obstáculos que han quedado desactualizados (en azul), para finalmente ser eliminados del sistema.

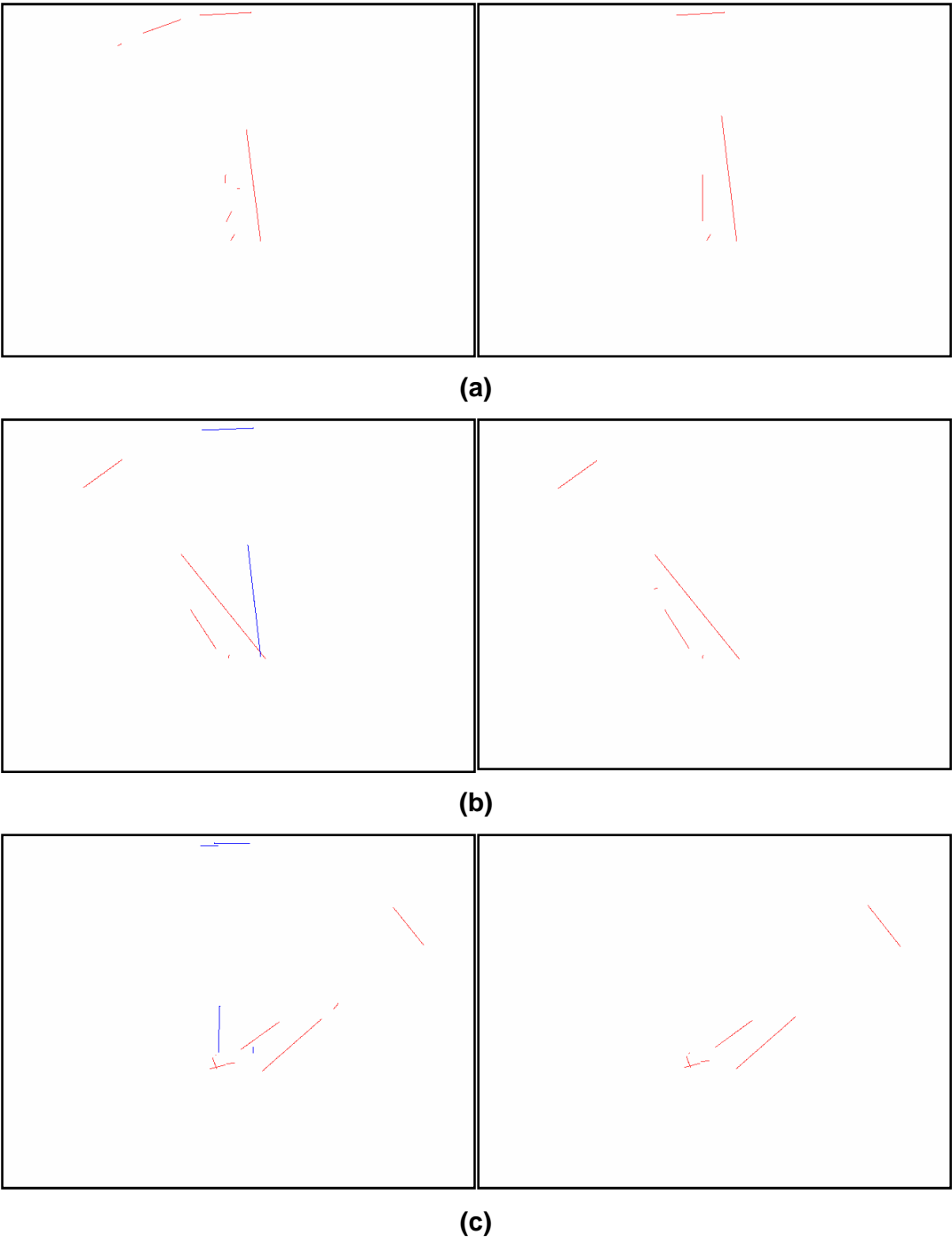


Figura 7-16: mapas de obstáculos con el láser (a) apuntando al frente, (b) girado a la derecha, (c) girado a la izquierda

Sistemas de referencia

Mención especial constituye el sistema de referencia necesario, que diferirá dependiendo de las necesidades y ámbitos que abarcan cada uno de los mapas mostrados.

Por un lado, los mapas que representan sólo el entorno próximo al móvil (I y II) deben emplear como sistema de referencia la posición y orientación del propio robot, por lo que no requerirán gran número de transformaciones ya que la información del láser se obtiene en función de la posición donde esté situado el coche.

Por otra parte, en el caso de reflejar una evolución en el movimiento (tanto si se trata de un modelo incremental – IV y V – como si no –III-), el mapa local estará definido mediante un sistema de referencia global cuyo centro se sitúa en la posición inicial del robot. A medida que este avanza, la configuración adoptada por el robot plantea la necesidad de un sistema de referencia local (lo que implica cambios en origen de coordenadas para la comparación de los distintos planos). Este asunto plantea dificultades en lo referente, principalmente, al ángulo de apuntamiento al que está orientado el móvil.

La Figura 7-17 muestra distintas mapas construidos a medida que se mueve el robot, donde se reflejan tanto la traslación como la rotación del mismo.

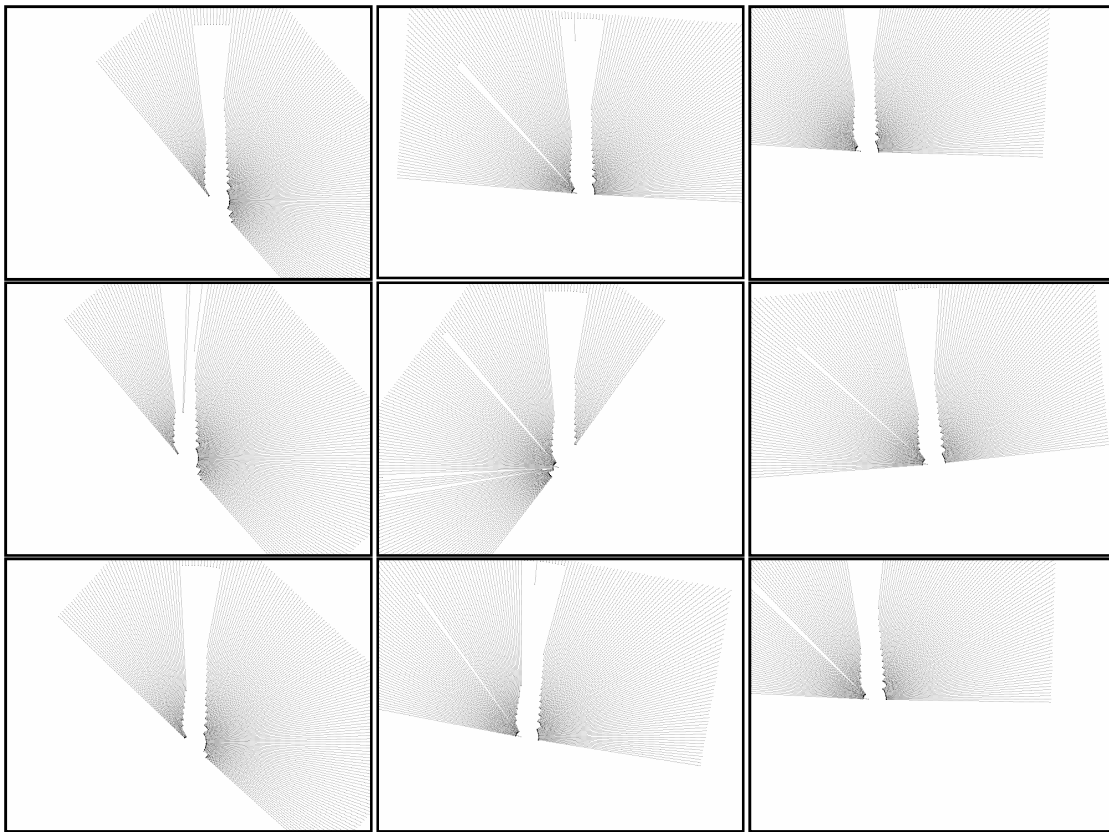


Figura 7-17: mapas de movimiento en función de la orientación del robot

7.4.2 Planificación local

El planificador proporciona al vehículo móvil las posiciones por las que debe pasar obligatoriamente para seguir el camino marcado a la meta. Sin embargo, no proporciona información relacionada con la trayectoria a seguir entre dos puntos, ya que dicho comportamiento es modelado mediante el uso de un planificador local. La razón principal de esta separación entre las funciones de planificación de ruta y seguimiento de la misma se fundamenta en la falta de precisión del mapa global, que le lleva a excluir del mismo los obstáculos fijos de menor tamaño o aquellos que se presentan de forma circunstancial, así como a la presencia de obstáculos móviles que irán modificando su posición a medida que el vehículo avance en su recorrido.

Algoritmo de búsqueda local

Al igual que en el caso del mapa global, se emplea el algoritmo proporcionado por las bibliotecas MRPT, donde tanto el móvil como las posiciones del camino a las que debe dirigirse (y por tanto, ocupar) son modelados como una circunferencia de radio R_p . Debido a las diferencias del espacio de búsqueda global y local, fue necesario modificar el valor a $R_p=1.5m$ ($R_p < R_G$), ya que con un valor mayor, no era posible obtener el camino aun existiendo una ruta libre de obstáculos entre las dos posiciones que había que conectar; principalmente, los problemas surgían cuando entorno a las posiciones inicial y final existían obstáculos próximos.

Localización local del vehículo

La búsqueda local parte de la primera posición donde se encuentra el robot. Debido a problemas de imprecisión en la localización (obtenida mediante señal GPS, capítulo 5.2) es necesario un ajuste previo para asegurar que esa posición inicial entra dentro de un margen correcto (en la práctica, la señal de posicionamiento muchas veces coloca el móvil en una zona del espacio incorrecta, incluso allí donde existen obstáculos). El criterio seguido es asegurar que el vehículo se encuentre situado en un entorno cercado próximo a la posición inicial del camino, cuyo valor es conocido, al ser empleado en la planificación global. Dicho planificador global también proporciona los puntos consecutivos a los que debe dirigirse el móvil (camino global); dependiendo de la configuración elegida se dispone de un número de posiciones mayor o menor dentro del alcance del robot (dicha cantidad depende del valor de fiabilidad que impongamos al entorno medido, ya que el láser proporciona un alcance de hasta 80 metros desde el punto donde se halle el robot, pero al aumentar la distancia de existencia de un posible obstáculo aumenta la incertidumbre sobre la integridad/fiabilidad del mismo)

Percepción del entorno local

Con la información obtenida mediante láser, se modela un esquema del entorno próximo al móvil, donde son señaladas las submetas a alcanzar (los puntos de la ruta) dentro de los límites del alcance prefijado. No es conveniente el empleo de un único punto de destino relativo o submeta, ya que aunque las posiciones a alcanzar

pueden considerarse de forma independiente, los movimientos del robot para llegar a ellas no lo son; es decir, las decisiones que tome el móvil deben encaminarle a recorrer el mejor trayecto que pase por los distintos puntos, y no orientarle únicamente a uno en concreto.

Condición de alcance de meta

Aunque el móvil debe dirigirse a un punto se considerará que ha llegado a su objetivo cuando se encuentre a una distancia cercana a esa circunferencia. La distancia mínima para considerar a la submeta como alcanzada es un parámetro ajustable, que en el caso práctico hemos impuesto a $3R_P$.

Los movimientos del robot van a diferir del cálculo teórico al desarrollo práctico. Esta razón, unida a desvíos en la posición, puede provocar que el móvil rebase la zona considerada como objetivo, teniendo que producirse un retroceso para su alcance. Con la idea de evitar esta situación, y sabiendo que existen otras metas posteriores del camino, la solución radica en tomar el siguiente punto del mismo como nuevo objetivo. Para ello, volvemos a la condición de distancia mínima a objetivos, comprobando no sólo que el móvil se encuentre próximo a la submeta buscada, sino también en un radio cercano al siguiente punto (dicho valor es ajustado a $6R_P$, el doble de la primera condición) para realizar este cambio de objetivos. El algoritmo empleado se muestra en la Figura 7-18.

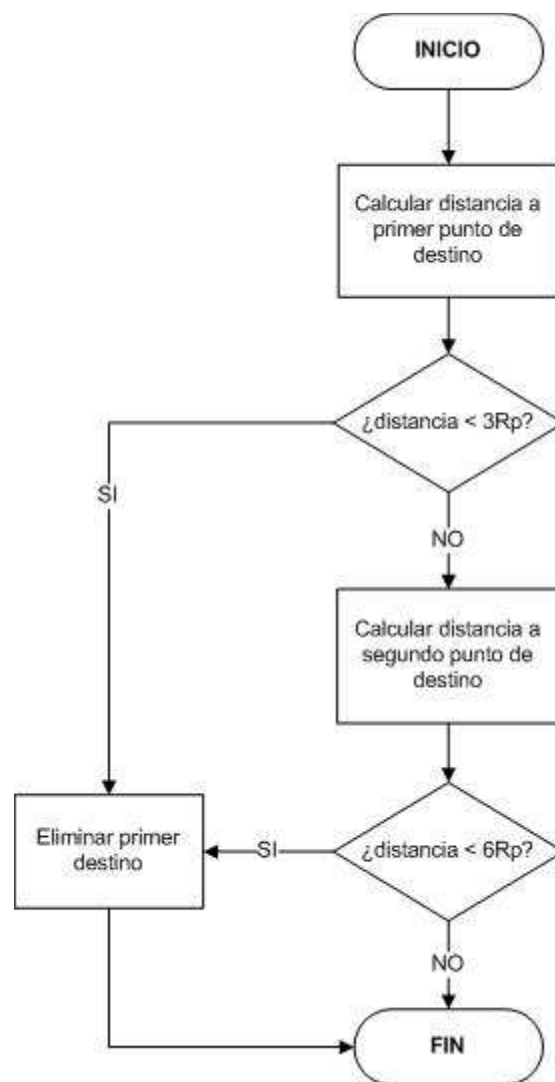


Figura 7-18: esquema de alcance de submeta

Problemas del camino local

De forma teórica, el espacio existente entre dos puntos consecutivos debería estar lo suficientemente despejado de obstáculos como para que el móvil pudiese desplazarse entre ellos. Sin embargo, pueden presentarse una serie de casos que constituyen excepciones:

i. Punto de destino dentro de la región de incertidumbre del mapa. Se produce cuando un obstáculo impide la visión directa del objetivo por hallarse en el camino que separa el móvil del mismo. Suele tratarse de un obstáculo fijo temporal o de un

elemento móvil, de dimensiones reducidas en comparación con los obstáculos fijos analizados por el planificador global. El objetivo se halla en una zona de la que actualmente carecemos de información, pero que presumiblemente se haya libre de obstáculos.

ii. Punto de destino en el interior o próximo a la zona concreta donde es detectado el obstáculo. Plantea mayores problemas que el caso anterior, ya que dicha proximidad indica cambios en el entorno no detectados de forma global. Si se trata de un obstáculo móvil, habría que detener el vehículo hasta que la posición quedase de nuevo libre. En caso de un obstáculo fijo, el acceso a dicho punto se hace inviable.

iii. Punto de destino fuera de los márgenes de barrido del láser. Son puntos que, aun situados dentro del alcance del vehículo, no se encuentran en el ámbito de visibilidad del mismo debido a su apuntamiento (ángulo de orientación). No tenemos información local relativa al mismo ni a su entorno.

iv. Puntos de valor atípico. Son, por ejemplo, aquellos que proceden de una curva innecesaria dentro del camino (una especie de arista desigual en la ruta). Son consecuencias del algoritmo de planificación seleccionado, y en la mayor parte de los casos no aportan información relevante para la consecución del camino. Se dan cuando aparece el vértice de un obstáculo angular que intercepta el camino, y hace que el punto intermedio entre otros dos puntos de recorrido más lineal se encuentre alejado de ambos, formando la arista en el camino antes mencionada.

En el caso (i), es necesario el análisis de las marcas posteriores a aquella que se encuentra oculta a la visión del vehículo. Si la siguiente marca se haya visible y accesible, una opción sería tomarla directamente como el siguiente punto de alcance e ignorar la marca oculta. En caso de una dependencia entre dicho objetivo posterior con el actual, seguramente no tendremos marcas visibles al alcance. El planificador debe entonces realizar una aproximación a la zona de incertidumbre y nuevamente ir comprobando la accesibilidad al objetivo. Seguiremos considerando, pues, que dicho destino es viable.

En el caso (ii) y en presencia de un obstáculo fijo que impida el acceso a la submeta siguiente, dicho objetivo se hace inviable y será necesario tomar la posterior posición en la ruta como submeta a alcanzar. Probablemente, la visión de dicho punto

subsiguiente se halle bloqueada por el obstáculo en cuestión, por lo que se llevarán a cabo las consideraciones del caso (i).

En el caso (iii), tenemos el vehículo apuntando a una dirección que no es la adecuada para alcanzar el siguiente punto del camino. Primero se comprobarán puntos posteriores de la ruta para evaluar la posibilidad de obviar el que se encuentra fuera del alcance del móvil. Si dichos puntos siguen fuera del alcance de visión, es necesario modificar el ángulo al que apunta el móvil para orientarlo en la dirección correcta (dirección del siguiente objetivo). Se puede hacer de diversas formas: modificando el ángulo y haciendo que el móvil avance cambiando su posición a partir de la que tenga actualmente; o realizando previamente un retroceso antes de avanzar para evitar que pierda la referencia al camino marcado (para que no avance demasiado y se pierda; en esta parte, puede ser muy útil el mapeado incremental, ya que nos aporta información del camino seguido hasta entonces y, por tanto, dónde están los obstáculos que se han ido dejando atrás). Este tipo de modificaciones no son contempladas en el algoritmo de planificación de caminos, ya que corresponde al ámbito de cálculo de trayectorias.

En el caso (iv) tenemos una arista de dimensiones reducidas que provoca una desviación en el camino. Si las dimensiones del móvil lo permiten, se podría ignorar dicha marca y seguir hasta la siguiente sin provocar la colisión del vehículo. Sin embargo, este razonamiento conlleva un riesgo elevado por la falta de precisión de los distintos elementos involucrados (posición exacta del vehículo, del obstáculo y del punto objetivo). Lo más adecuado sería rodear el obstáculo en cuestión mediante el empleo de un algoritmo de seguimiento de frontera, como pueden ser los de tipo BUG [Lop08-rea] [Lop08-sen]. Al igual que en el caso anterior (caso iii), esta situación se encuentra fuera del alcance de este proyecto.

La solución implementada en la práctica pretende solventar (si ello es posible) las dificultades relacionadas con la posible inaccesibilidad de las submetas a alcanzar, centrando su atención en los casos (i) y (ii) anteriormente expuestos. La solución radica en contemplar más de un objetivo a la hora de calcular el camino local (ver Figura 7-19), que por simplicidad en los cálculos hemos reducido a tres posiciones consecutivas del camino global (calculado previamente en el apartado 7.3.2).

La idea fundamental es que el planificador local siempre devuelva un posible camino, ya sea real (libre de obstáculos) o de aproximación a una zona ocupada; de esta forma, el algoritmo busca la existencia de camino libre para alguna de las posiciones, y en caso de no encontrarlo, devuelve el camino de aproximación a la primera submeta (aunque este no sea considerado como libre). Ello permite que el robot continúe su avance, ofreciendo la posibilidad de encontrar posteriormente el verdadero camino libre de obstáculos una vez las condiciones del medio (y las suyas propias) hayan cambiado.

Por último, señalar que el algoritmo de búsqueda local está diseñado para calcular el camino cuando el robot avanza por su entorno, no cuando retrocede. Por ello, si el sentido de movimiento encamina al móvil hacia posiciones ya pasadas (vuelta atrás en el mapa), el comportamiento de la aplicación no será el esperado, produciendo errores en el cálculo del camino.

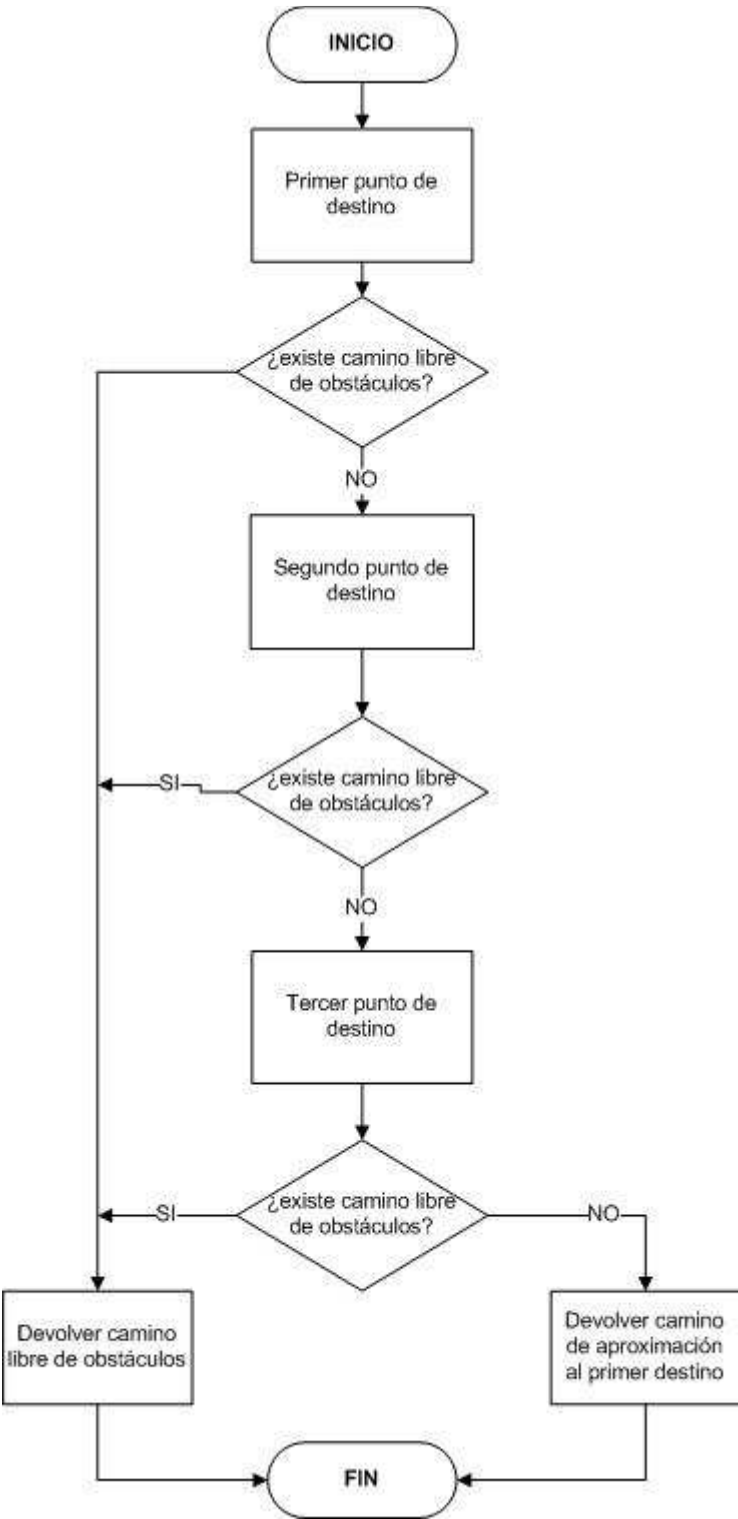


Figura 7-19: esquema del tratamiento de las posiciones a alcanzar

7.5 Mapeado incremental mediante el modelo de obstáculos

Cada vez que se recibe una nueva actualización de los datos, los mapas locales van cambiando en función de los obstáculos que aparecen. Un mapa que sea de tipo incremental actúa de referente para comprobar cómo se va actualizando dicha información, es decir, si los obstáculos encontrados acaban de presentarse en el campo visual del móvil o si se corresponden en mayor o menor medida con los detectados anteriormente, tal y como se explicó en el apartado 4.2.3.

El punto de partida es un mapa de rejillas de ocupación, en el que serán representados tanto los obstáculos nuevos detectados como los que han sido observados anteriormente. El tratamiento e interpretación de la información discreta obtenida por el láser genera un conjunto de estructuras geométricas, luego podemos considerar el mapa así construido como una representación jerárquica o híbrida en dos niveles: en el nivel inferior se dispone la información mediante un modelado en rejilla, mientras que en el superior se sitúan los datos geométricos correspondientes [Gal07].

De esta forma, se va actualizando la base de conocimientos del robot entre iteraciones sucesivas, con acceso a los datos almacenados. Esto sirve sobre todo para reconocer:

- Obstáculos móviles, ya que entre iteraciones consecutivas se consideran como elementos distintos.
- Errores producidos en alguna de las tomas de datos, ya que todo se almacena y compara.

7.5.1 Incorporación de los obstáculos

La posición en la que se encuentran los obstáculos es fundamental para el correcto encaminamiento del robot. No sólo es necesario saber dónde ir, sino también si es posible alcanzar esa posición en las condiciones actuales. No olvidemos que la información global del entorno puede ser cambiante debido a la aparición de

obstáculos fijos no registrados: vehículos parados, montículos, bancos... Además, por supuesto, de la aparición de obstáculos móviles cuya presencia es necesario tratar con especial atención. En la práctica, el sistema se centra en el modelado de los obstáculos fijos en un entorno próximo al móvil.

Cuando el vehículo experimenta un desplazamiento, los obstáculos observados hasta ese momento también cambian su posición en relación al móvil (que es siempre la referencia a seguir en el algoritmo de mapeado). Ese cambio de coordenadas y, sobre todo, de orientación (ángulo) al que apunta el móvil dificulta la comparación de objetos detectados a la hora de clasificarlos como un mismo obstáculo o como obstáculos distintos.

Los datos del láser, aunque obtenidos respecto a la posición del robot, son transformados al sistema absoluto de coordenadas para su representación en el mapa incremental, por tanto sobre dichos valores no se realizará ningún otro cambio de posición para su tratamiento.

Los datos del láser son procesados, agrupándolos y realizando un ajuste de los mismos mediante segmentos lineales. Posteriormente, cada segmento es interpretado como un objeto independiente que recibe el nombre de obstáculo.

Cada objeto obstáculo es una estructura que se identifica con dos puntos, representando el inicio y el fin del segmento modelado; así mismo, para cálculos posteriores, se calcula su punto medio. Los obstáculos serán almacenados en dos listas enlazadas: una con la información actual, y otra con la obtenida hasta el momento.

Cuando se realiza una nueva medición, los obstáculos contruidos se almacenan y comparan con los ya existentes en la memoria del robot, y se procede a la eliminación de datos obsoletos y a la actualización de los nuevos. Cabe destacar que la información del instante actual tiene preferencia sobre la pasada, ya que es preciso que el sistema de modelado de obstáculos reaccione ante cambios en el entorno.

De esta forma, el proceso de actualización consta de los siguientes pasos:

- Eliminación de datos no válidos actualmente
- Búsqueda de obstáculos anteriores cercanos al nuevo
- Comparación de elementos, y en caso de éxito, actualización del obstáculo nuevo y eliminación del antiguo
- Integración de la información en memoria

La Figura 7-20 representa los mapas creados en diferentes momentos del avance del robot. En color rojo, se muestran los obstáculos actuales obtenidos a partir de la información captada por el láser en cada instante; en color azul, los obstáculos anteriormente modelados que permanecen en memoria, gran parte de los cuales serán posteriormente eliminados al tratarse de información obsoleta. Se puede comprobar que algunas de las imágenes muestran un esquema no válido del entorno, con gran acumulación de obstáculos históricos; a medida que avanza el algoritmo, la situación se estabiliza y tienden a prevalecer los valores reales.

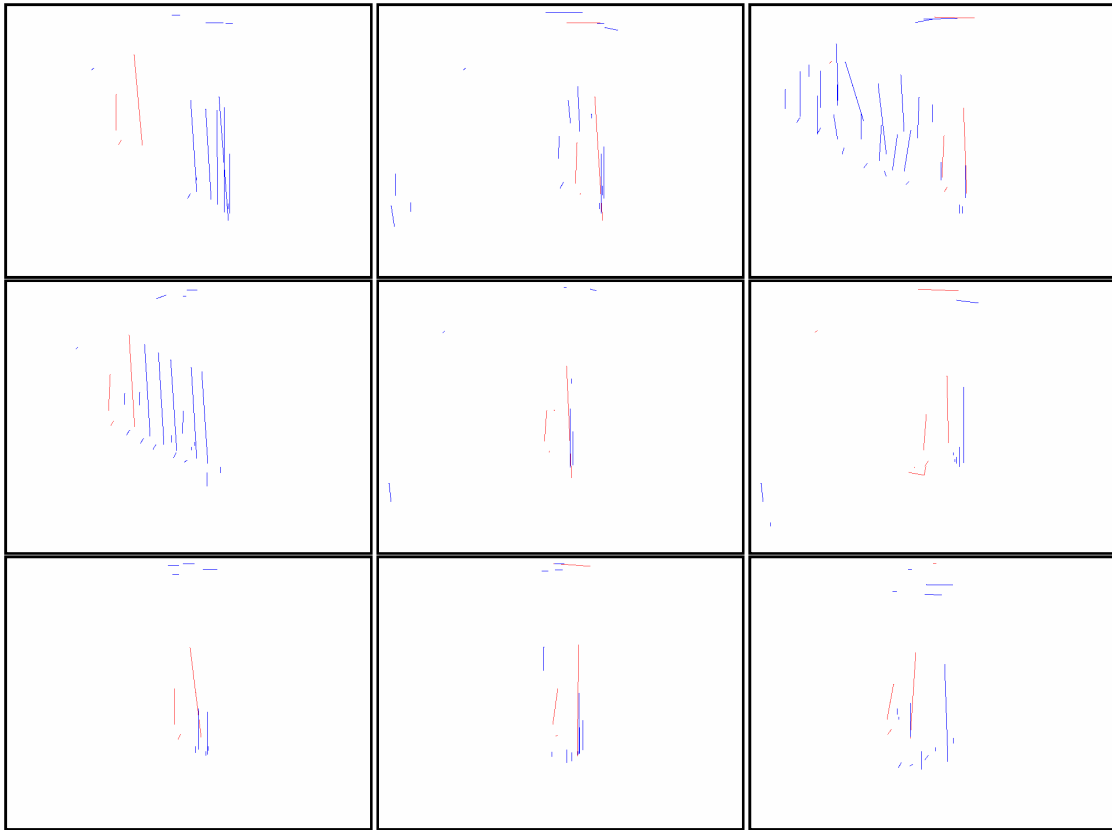


Figura 7-20: mapas de obstáculos en diferentes momentos del avance del robot

Eliminación de información obsoleta

El primer paso de la actualización consiste en el tratamiento de los obstáculos no válidos actualmente. Para ello, se recurre a la información del láser, que nos proporciona el área libre de colisiones. Los obstáculos anteriores que se encuentren dentro del área de libre circulación son recortados en función de la intersección de los rayos láser sobre ellos, y en caso de una reducción significativa (fijada por parámetro), eliminados totalmente de memoria (ver Figura 7-21).

Tras este proceso, sólo quedan almacenados los elementos que se encuentren en zona de colisión, que serán comparados con los objetos nuevos con el fin de buscar las correspondencias existentes.

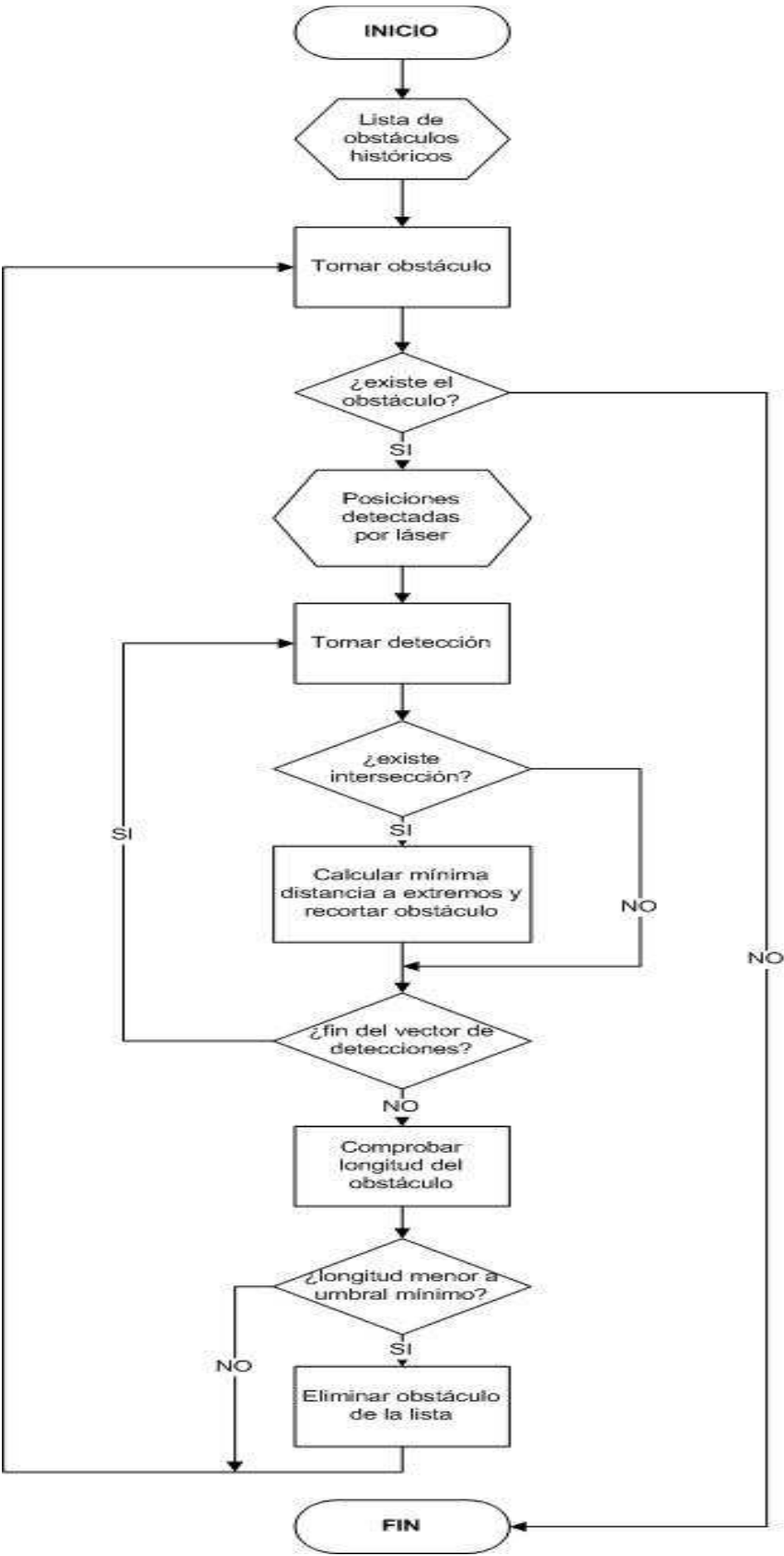


Figura 7-21: esquema de eliminación de obstáculos

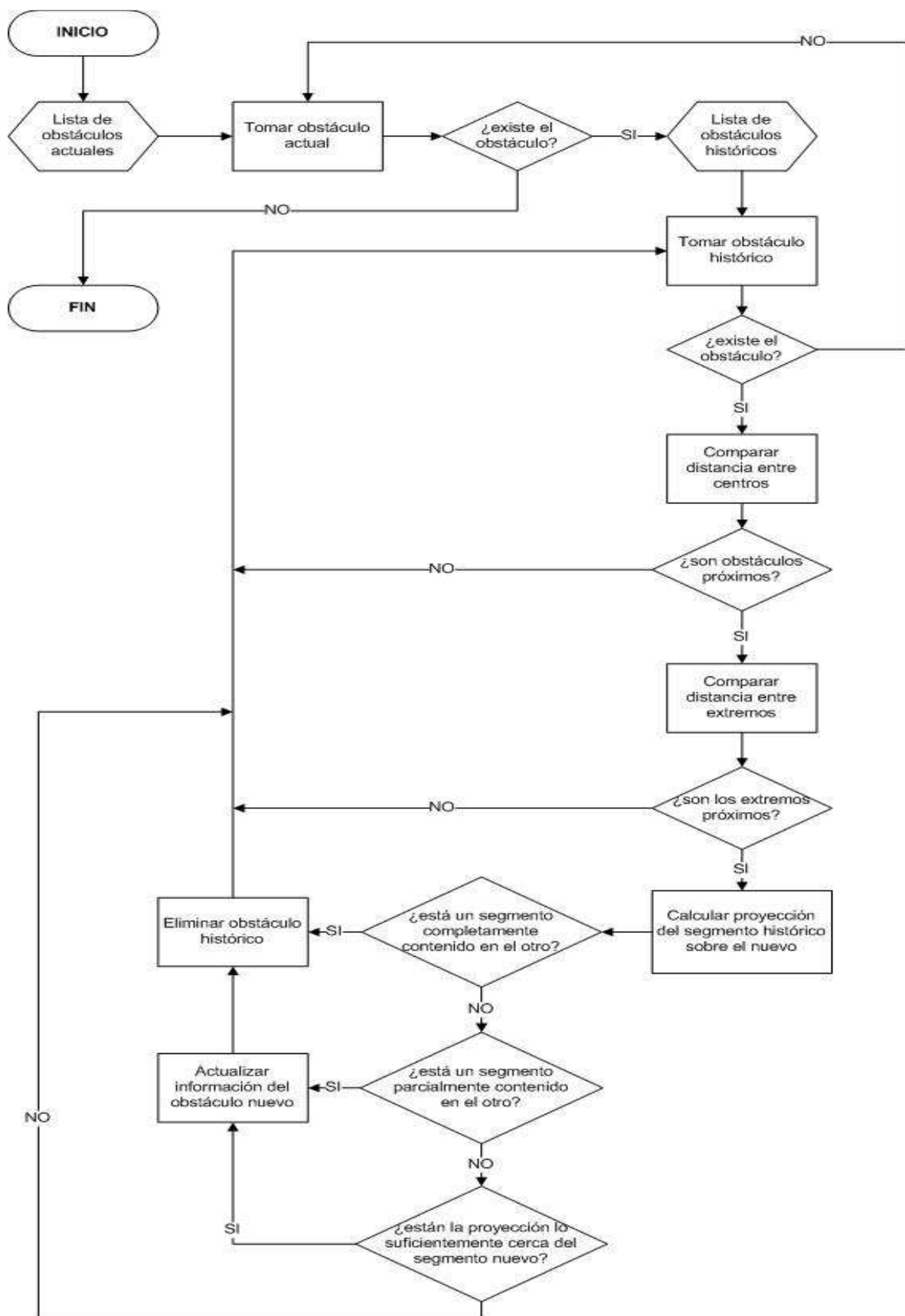


Figura 7-22: esquema de actualización de obstáculos

Búsqueda de obstáculos próximos

Cada obstáculo nuevo será comparado con todos aquellos existentes en memoria que se encuentren próximos a él (ver Figura 7-22). La condición de cercanía se modela mediante la distancia euclídea entre centros, que debe ser menor a un determinado valor; en la práctica, fijado a 5 metros.

Comparación de obstáculos

Cuando un obstáculo antiguo es considerado como próximo a uno nuevo, se pueden dar dos circunstancias: que sean dos objetos independientes, o que exista una relación entre ellos. En el primero de los casos, el elemento anterior se conserva para posteriores comparaciones; en el segundo, la información antigua es integrada en el nuevo objeto. El proceso completo se encuentra representado en la Figura 7-22.

La primera comprobación a efectuar consiste en calcular las distancias de los extremos del segmento antiguo a la recta definida por el segmento nuevo (o dicho de otra manera, la distancia entre cada extremo y su proyección sobre la recta). Si dichas posiciones se encuentran a una distancia menor a una dada, sobre el objeto antiguo se efectuará una segunda tanda de comprobaciones; en caso contrario, lo consideramos como independiente y lo mantenemos en memoria.

La segunda tanda de comprobaciones se basa en la búsqueda de relaciones entre ambos segmentos. Para ello se comprueba la posible intersección entre el segmento nuevo y la proyección del antiguo, pudiendo darse los siguientes casos:

- La proyección se haya contenida completamente dentro del segmento nuevo, por lo tanto consideramos que se trata del mismo elemento. El segmento antiguo no aporta información nueva que repercuta en el momento presente, por lo que el obstáculo antiguo es eliminado sin afectar al obstáculo nuevo.
- La proyección se haya parcialmente contenida dentro del segmento nuevo, por lo tanto consideramos que se trata del mismo elemento (aunque con variaciones). La información del segmento nuevo es actualizada con la del antiguo, el cual es posteriormente eliminado.

- La proyección se haya totalmente fuera del segmento nuevo, por lo que puede tratarse del mismo obstáculo o de otro distinto. Aquí es necesario otra comprobación de distancias, esta vez para saber cómo de lejos está la proyección (no el segmento) respecto al obstáculo nuevo. Si dicha distancia es menor que un valor dado, se considera que ambos objetos son el mismo, se actualiza la información del nuevo y se elimina el antiguo; en caso contrario, se considera el elemento antiguo como independiente, y los mantenemos en memoria (sin modificar el obstáculo nuevo).

Integración en memoria

Una vez finalizado el proceso para todos y cada uno de los obstáculos nuevos, la información referente a los mismos (modificada o no), junto a la relativa a los obstáculos antiguos considerados independientes, es almacenada en memoria, constituyendo la base de comparación en la siguiente iteración.

Justificación del proceso de actualización

El proceso de interpretación de los obstáculos requiere el empleo de una lógica aplicada al tratamiento de datos, formada por la secuencia de operaciones matemáticas descritas; tras la aplicación de cada uno de los pasos anteriores, se va cerrando la incertidumbre sobre la naturaleza de los obstáculos existentes.

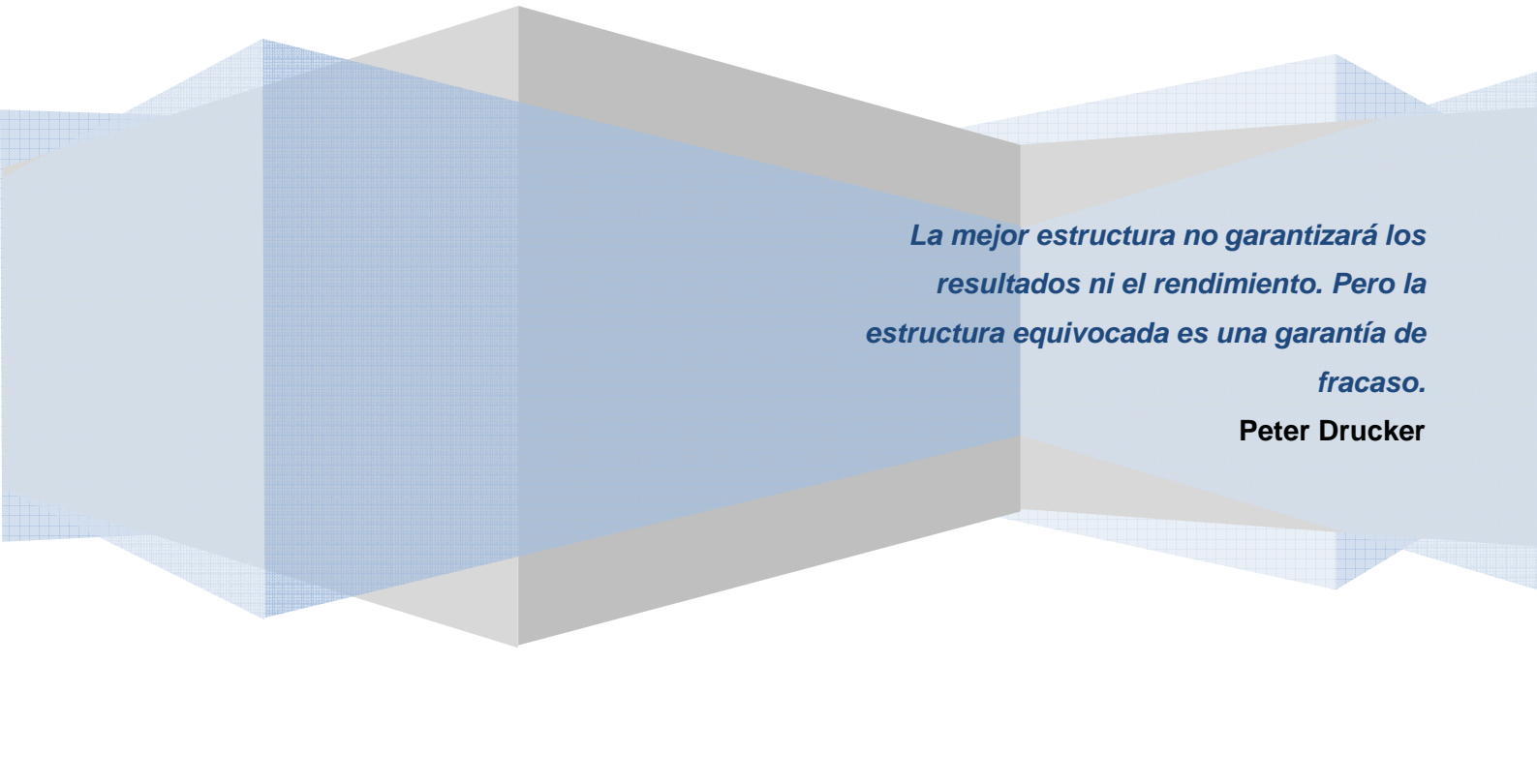
El objetivo de la fase de eliminación radica en que no deben haber obstáculos en la zona actual de libre tránsito; por tanto, da preferencia a los obstáculos situados en las zonas de colisión e incertidumbre, lo que no asegura que los datos resultantes de la operación tengan validez en el momento actual, sólo que se encuentran en el lugar adecuado. De forma ideal, el barrido realizado por el láser debería limpiar el espacio libre que rodea al móvil; sin embargo, debido a la naturaleza discreta de los haces, es posible la permanencia de elementos si éstos se encuentran situados entre dos valores de medición (se trata de obstáculos verticales de escasa o nula pendiente), sobre todo a medida que nos alejamos de la posición actual del móvil. Aunque en un primer momento esta situación pudiera inducir a error, a medida que

continúa la actualización resulta cada vez más probable su eliminación (ya que se refina la búsqueda de intersecciones al aproximarse el láser al objeto)

En la comparación, se busca evitar actualizaciones incorrectas, considerándose preferente mantener un obstáculo anterior (aunque existan posibilidades de que corresponda a uno actual) a integrar información no válida, ya que dicho error permanecería y se propagaría en posteriores iteraciones; mientras que manteniendo la información, es más probable que esta sea eliminada o actualizada en posteriores mediciones.

CAPÍTULO 8.

RESULTADOS



*La mejor estructura no garantizará los
resultados ni el rendimiento. Pero la
estructura equivocada es una garantía de
fracaso.*

Peter Drucker

8. Resultados

En este apartado, se muestran una serie de secuencias obtenidas mediante la realización de pruebas en el entorno de trabajo (explicado en el apartado 4.1). En dicho contexto, se realizaron una serie de simulaciones reales, tanto en situaciones estáticas (robot parado) como dinámicas (robot en movimiento), de las cuales se han seleccionado las más características y las que ofrecen mejores prestaciones.

Todas las simulaciones se han realizado empleando el mismo intervalo de tiempo, aunque el número y tipo de resultados se ha variado para tratar de mostrar distintos aspectos de la implementación.

8.1 Pruebas en estático

Las pruebas en estático comprenden las pruebas realizadas con el robot parado, por tanto no se realizan acciones de translación pero sí de rotación (cuando se considere necesario).

El objetivo de realizar estas pruebas consiste en tratar de identificar, de forma aislada, una serie de factores que influyen en el desarrollo del algoritmo de búsqueda local (lo que resulta más complicado ante cambios de posición del vehículo), estudiando la influencia de los mismos en el correcto funcionamiento de la aplicación; estos factores, de forma general, son:

- Robustez y fiabilidad de los datos proporcionados por láser.
- Capacidad de encontrar un camino libre de obstáculos en un entorno estático (que puede ser similar al empleado en búsqueda local, o presentar elementos diferentes).
- Influencia de los obstáculos móviles en el desarrollo del algoritmo.
- Efectos de realizar una rotación independientemente del cambio de posición.

- Influencia de la deriva en la posición y fiabilidad de los datos proporcionados por el GPS.

8.1.1 Simulación 1

La Figura 8-1 muestra una secuencia de mapas de movimiento (aunque estemos en un caso estático), donde se realiza la búsqueda del camino local conocidos los puntos a alcanzar o submetas del camino global (círculos azules). En caso de encontrar un camino libre de obstáculos, la ruta a seguir se representa en color rojo; en caso contrario, se obtiene un camino de aproximación al destino más próximo, que se representa en color naranja.

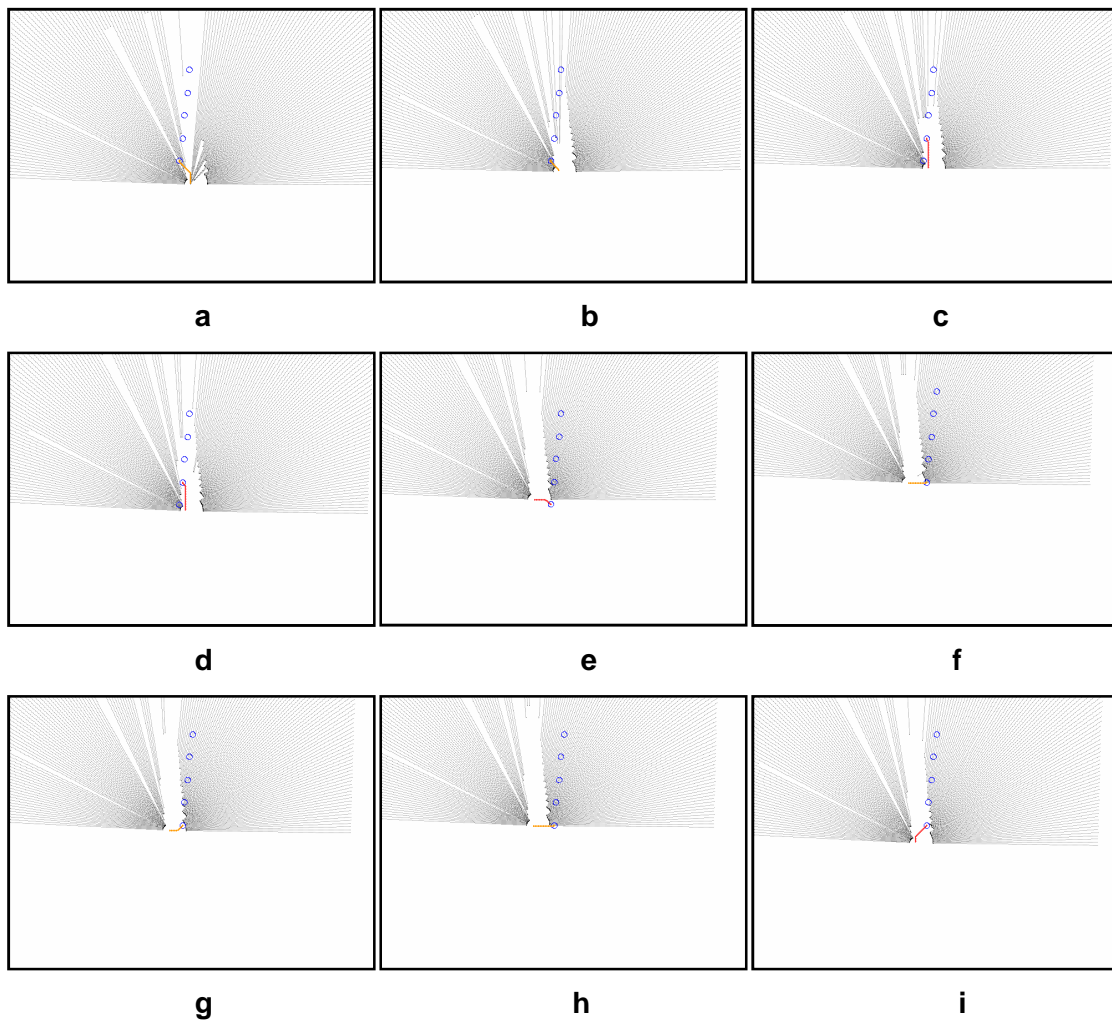


Figura 8-1: búsqueda local – caso estático (I)

Obstáculos móviles (a,b,c,d): al comienzo de la simulación, al algoritmo no es capaz de encontrar una ruta libre de obstáculos(a), debido a que en ese momento se produce el paso de peatones en una zona tan cercana al láser que bloquea toda posibilidad de camino. A medida que los peatones se alejan, el espacio libre se amplía hasta ser posible obtener un camino (c) a la segunda submeta, ya que la primera queda situada en la zona ocupada.

Deriva de la posición (e,f,g,h,i): se pierde la referencia de posición, y durante las siguientes medidas, no es posible encontrar el camino correcto, sólo una aproximación a la primera submeta. Finalmente, la posición tiende a estabilizarse a su valor real, y es posible el cálculo del camino (i).

La Figura 8-2 corresponde a varias muestras de mapas de obstáculos, tomadas a lo largo de la presente simulación; en rojo se representan los obstáculos actualmente medidos y, en azul, los históricos (obstáculos anteriores almacenados en memoria).

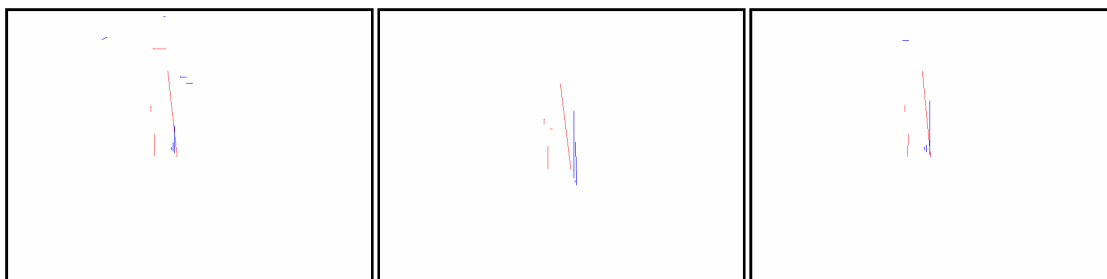


Figura 8-2: mapas de obstáculos – caso estático (I)

8.1.2 Simulación 2

La Figura 8-3 muestra una secuencia de mapas de movimiento, donde el camino libre de obstáculos se representa en rojo, y el de aproximación en naranja. En esta prueba se han tomado un número mayor de imágenes que en la simulación anterior, con el fin de tener una visión más amplia de los cambios que se producen durante el desarrollo del algoritmo.

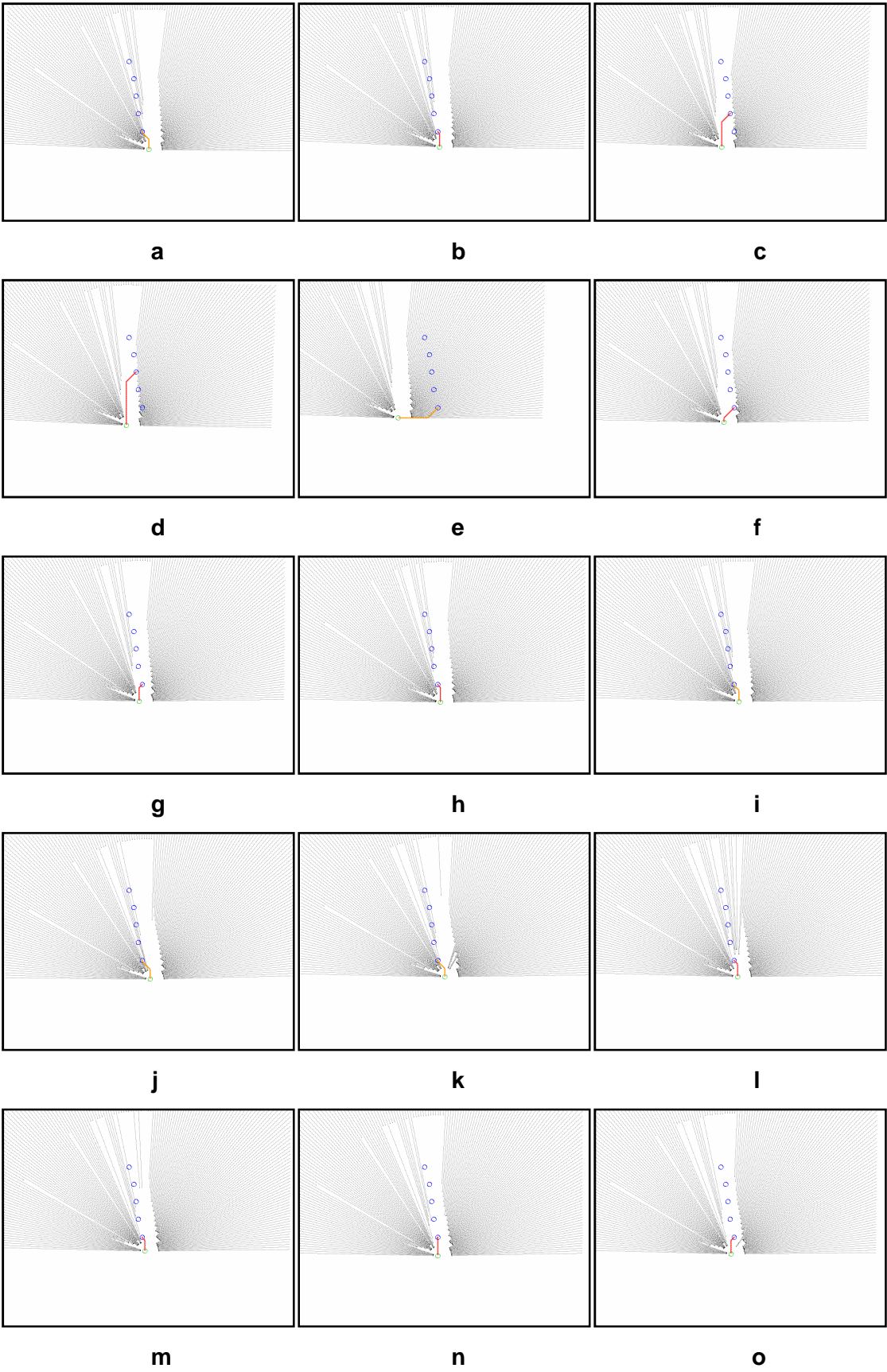


Figura 8-3: búsqueda local – caso estático (II)

Cambio de elección de meta (a,b,c,d): debido a la deriva de posición, los objetivos a alcanzar pasan de la primera submeta (b) a la segunda (c) y tercera (d), para cada una de las cuales se encuentra el camino correcto a cada una de ellas.

Perdida de la referencia de posición (e): debido a la deriva antes comentada.

Situación estable (f,g,h): se encuentra el camino que conduce a la primera submeta.

Problemas debido a la deriva (i,j): sin producirse un cambio brusco de posición, la deriva existente provoca la pérdida de visión de los objetivos, por lo que el algoritmo proporciona una aproximación a la primera submeta.

Obstáculos móviles (k,l,m): se puede comprobar la presencia de peatones y su evolución en el espacio de visión del robot, aunque no interfiere con el cálculo del camino.

Situación estable (n,o): el algoritmo devuelve el camino correcto a la primera submeta.

La Figura 8-4 corresponde a varias muestras de mapas de obstáculos, tomadas a lo largo de la presente simulación; en rojo se representan los obstáculos actuales y, en azul, los históricos.

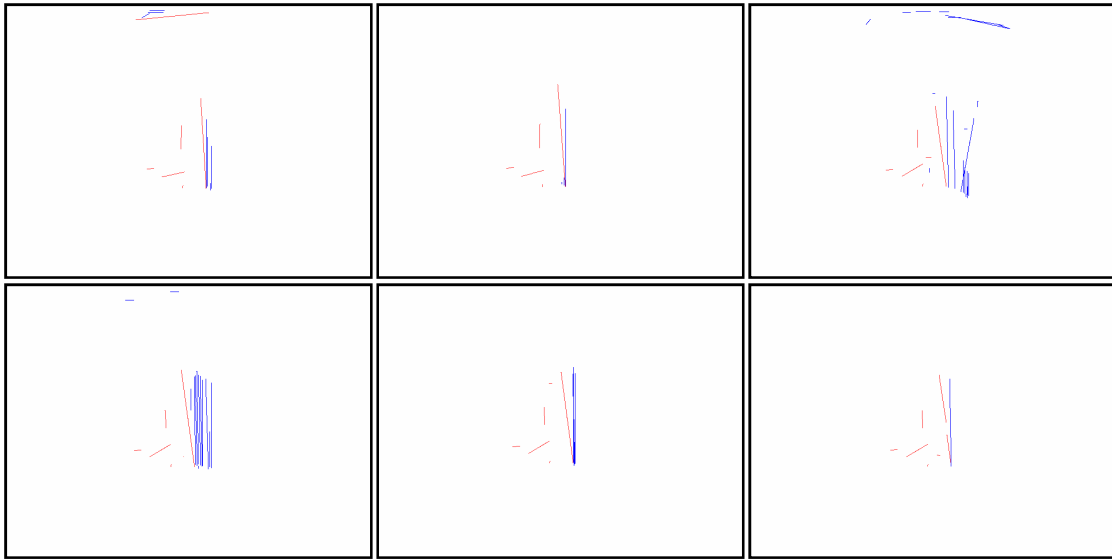


Figura 8-4: mapas de obstáculos – caso estático (II)

Figura 8-5 representa el mapa incremental obtenido tras la finalización de la presente simulación. Se pueden observar los efectos de variaciones en la posición debidas a la deriva del dispositivo GPS, lo que provoca saltos en las medidas tomadas y la obtención de una imagen poco homogénea.

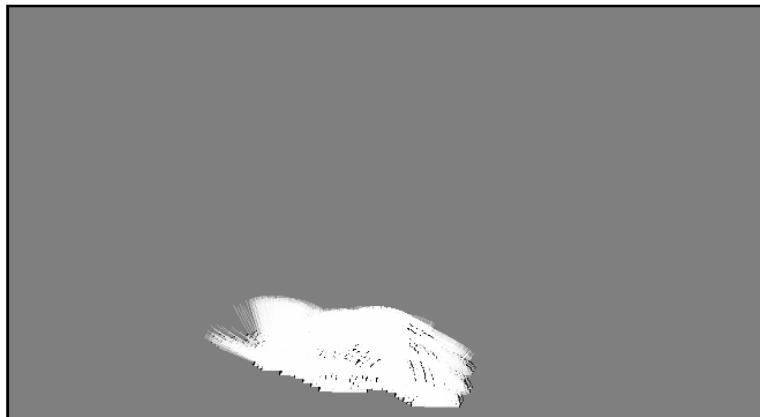


Figura 8-5: mapa incremental – caso estático (II)

8.1.3 Simulación 3

En esta prueba, se ha decidido tomar dos secuencias de mapas de movimiento: una devuelta por la aplicación de forma automática, donde cada imagen se obtiene tras un número fijo de iteraciones; y otra tomada por el propio usuario cuando se ha observado algún aspecto de interés. La primera secuencia se emplea para explicar el desarrollo de la aplicación, mientras que la segunda va a servir de respaldo a dichas explicaciones.

La Figura 8-6 muestra la secuencia automática de mapas de movimiento, donde el camino libre de obstáculos se representa en rojo, y el de aproximación en naranja.

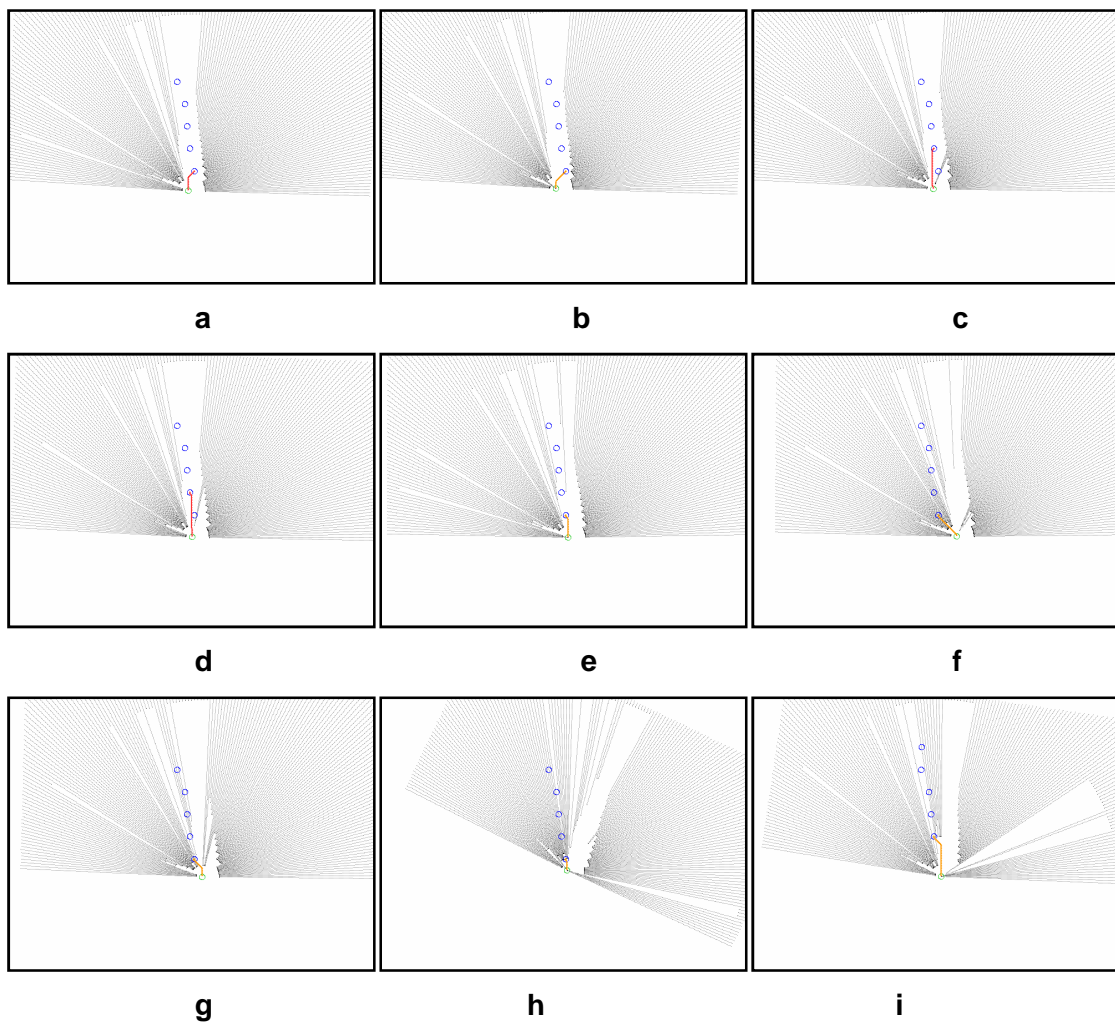


Figura 8-6: búsqueda local – caso estático (III)

Problemas debido a la deriva (a,b): el sistema pasa de encontrar el camino (a) a perderlo (b).

Primer obstáculo móvil (c,d,e): el paso de un peatón bloquea cualquier ruta a la primera submeta, por lo que el algoritmo devuelve el camino a la segunda (c,d); una vez el peatón se ha alejado lo suficiente, se obtiene el camino al primer objetivo (e).

Segundo obstáculo móvil y problemas debido a la deriva (f,g): aunque el paso de un nuevo peatón no afecta al algoritmo de búsqueda, sí lo hace la deriva en la posición.

Efecto de la rotación (h,i): se prueba a realizar un giro del robot, lo que permite encontrar de nuevo el camino (i).

La Figura 8-7 corresponde a los ejemplos de interés tomados durante la simulación. Se puede observar los efectos del paso del primer peatón (b,c,d), del segundo peatón (e), y la rotación del robot (f).

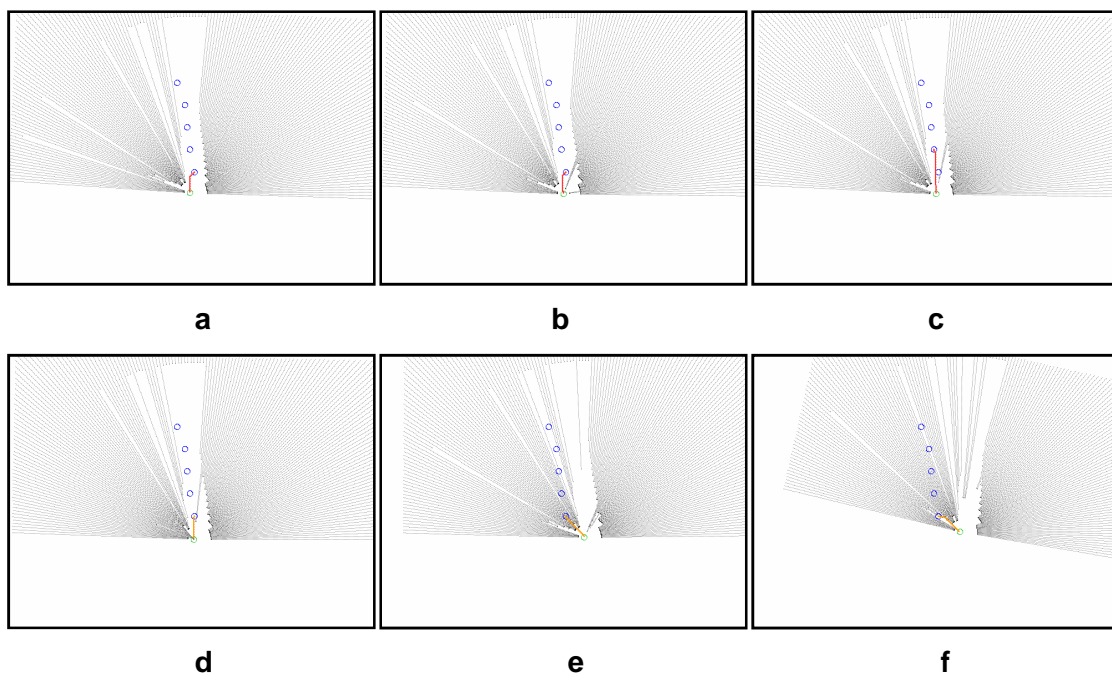


Figura 8-7: ejemplos extraídos de la búsqueda local – caso estático (III)

La Figura 8-8 corresponde a varias muestras de mapas de obstáculos, tomadas a lo largo de la presente simulación; en rojo se representan los obstáculos actuales y, en azul, los históricos. El aspecto más destacable corresponde a la quinta imagen de las seis que constituyen la secuencia, que corresponde a uno de los momentos en los que se realiza la rotación del robot (y donde la orientación de los obstáculos actuales difiere de la orientación de los históricos).

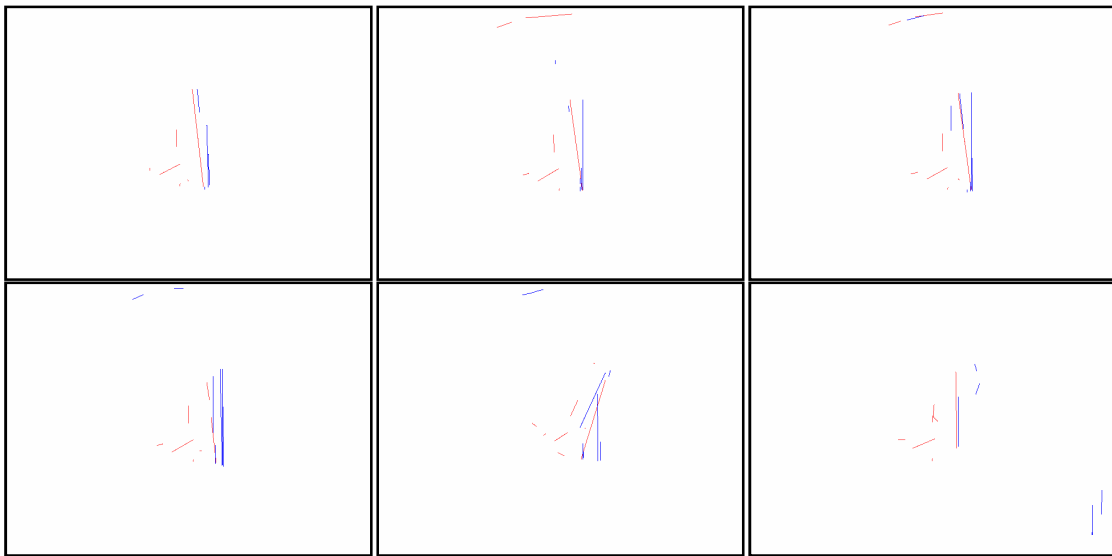


Figura 8-8: mapas de obstáculos – caso estático (III)

8.1.4 Simulación 4

Al igual que en la prueba anterior, en esta se ha decidido tomar dos secuencias de mapas de movimiento: una de forma automática y otra tomada por el propio usuario.

La Figura 8-9 muestra la secuencia automática de mapas de movimiento, donde el camino libre de obstáculos se representa en rojo, y el de aproximación en naranja.

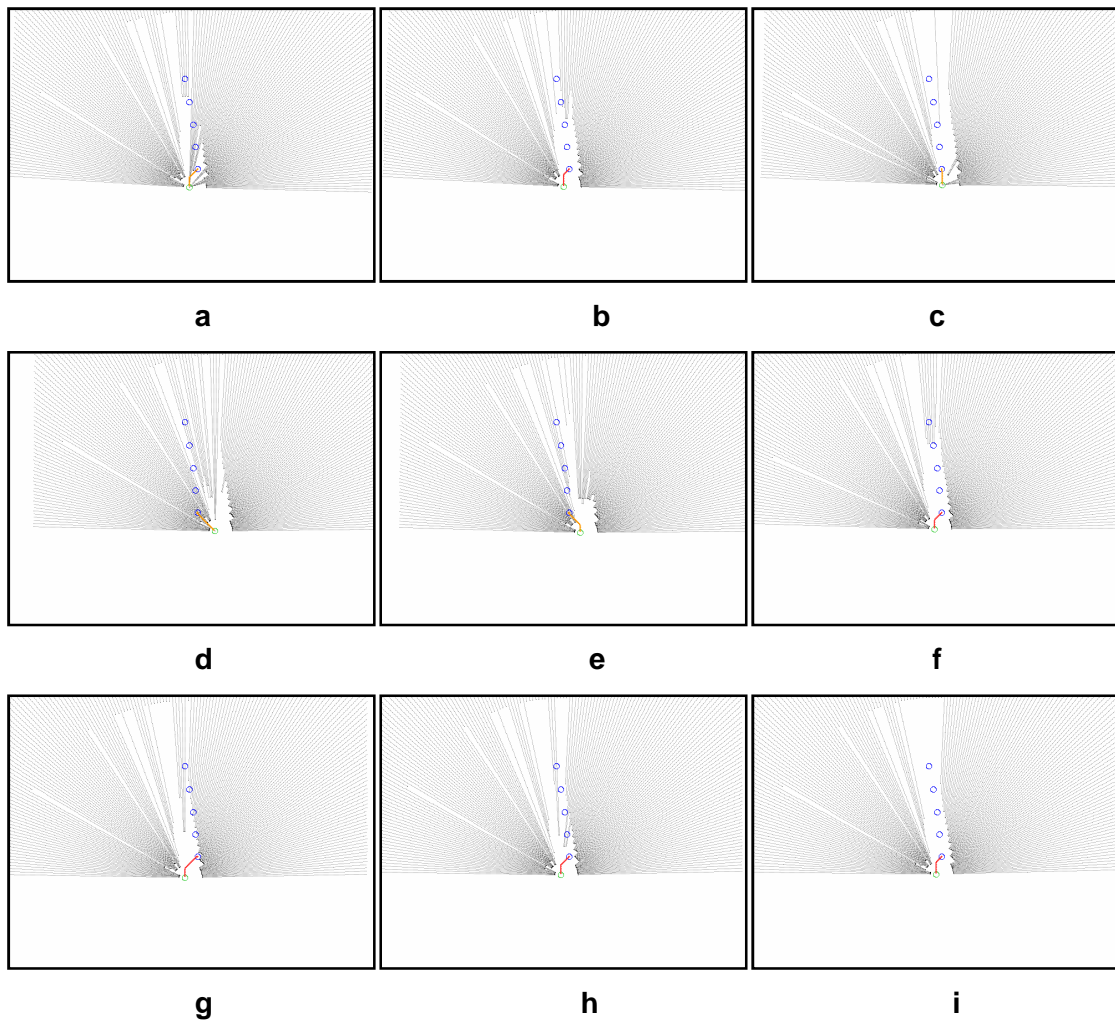


Figura 8-9: búsqueda local – caso estático (IV)

Primer obstáculo móvil (a,b): la existencia de peatones impide encontrar un camino libre (a) debido a su proximidad al láser; una vez que se han alejado del mismo, se obtiene el camino a la primera submeta (b).

Segundo obstáculo móvil y problemas debido a la deriva (c,d,e,f): ocurre igual que en el caso anterior, pero esta vez se suma el problema de falta de precisión en la posición; una vez que existe suficiente espacio libre frente al robot, y cuando la posición se estabiliza a la real, el algoritmo encuentra el camino (f).

Tercer obstáculo móvil (g,h,i): no afecta a las muestras tomadas, obteniéndose el camino a la primera submeta.

La Figura 8-10 corresponde a los ejemplos de interés tomados durante la simulación; en ellos se puede observar los efectos del paso de peatones en la obtención del camino local.

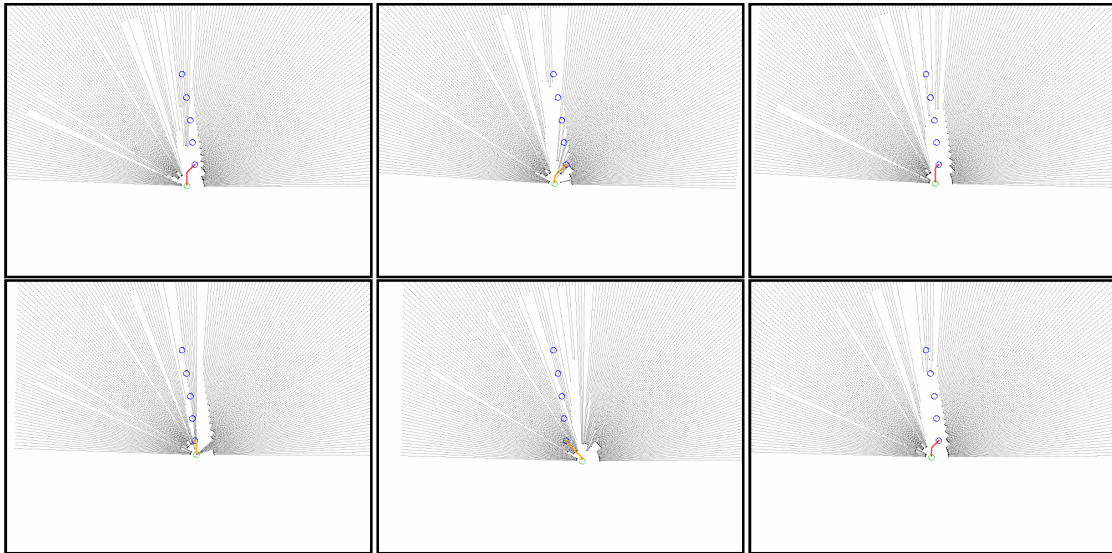


Figura 8-10: ejemplos extraídos de la búsqueda local – caso estático (IV)

La Figura 8-11 corresponde a varias muestras de mapas de obstáculos, tomadas a lo largo de la presente simulación; en rojo se representan los obstáculos actuales y, en azul, los históricos. En la cuarta figura de la secuencia se observa una acumulación de datos históricos en la parte derecha, que posteriormente serán actualizados y, en caso necesario, eliminados de memoria.

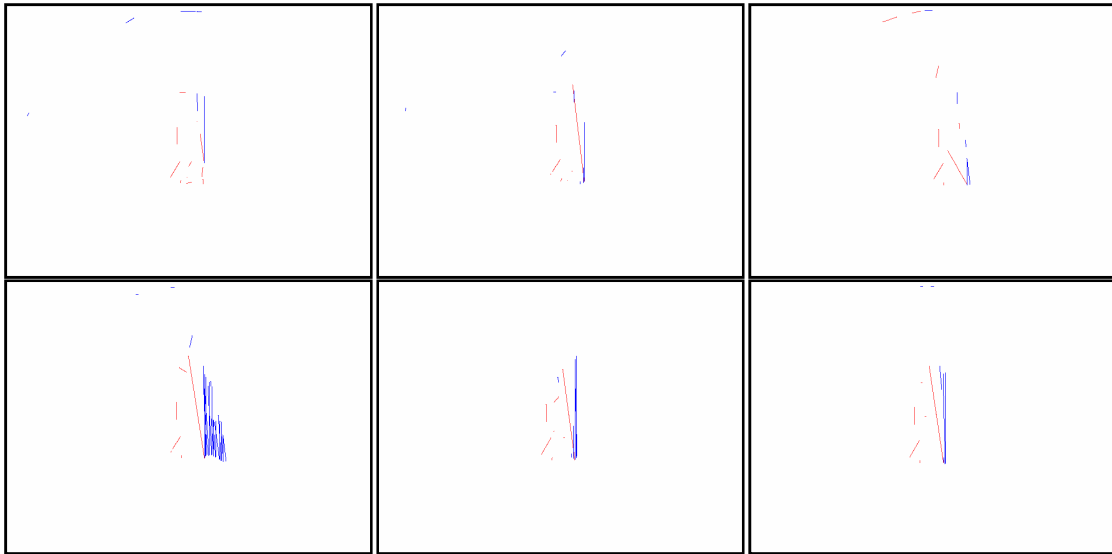


Figura 8-11: mapas de obstáculos – caso estático (IV)

La Figura 8-12 representa el mapa incremental obtenido tras la finalización de la presente simulación. Se puede observar que la deriva no afecta en exceso al mapa final, debido a que en esta simulación la posición del vehículo ha contado con una estabilidad mayor que en anteriores pruebas; por tanto, se trata de medidas estables y con alto grado de fiabilidad.

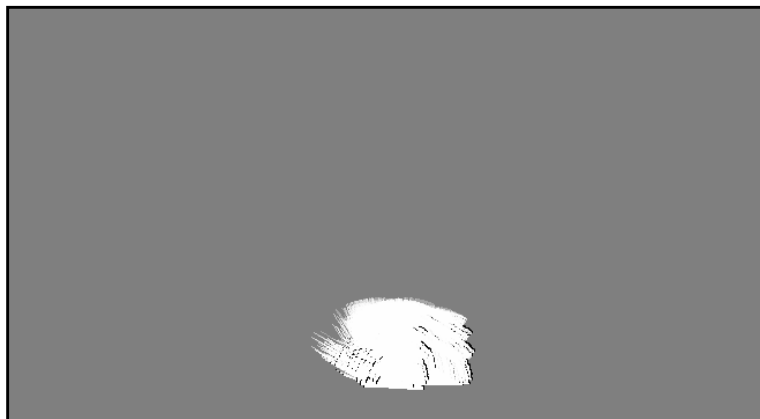


Figura 8-12: mapa incremental – caso estático (IV)

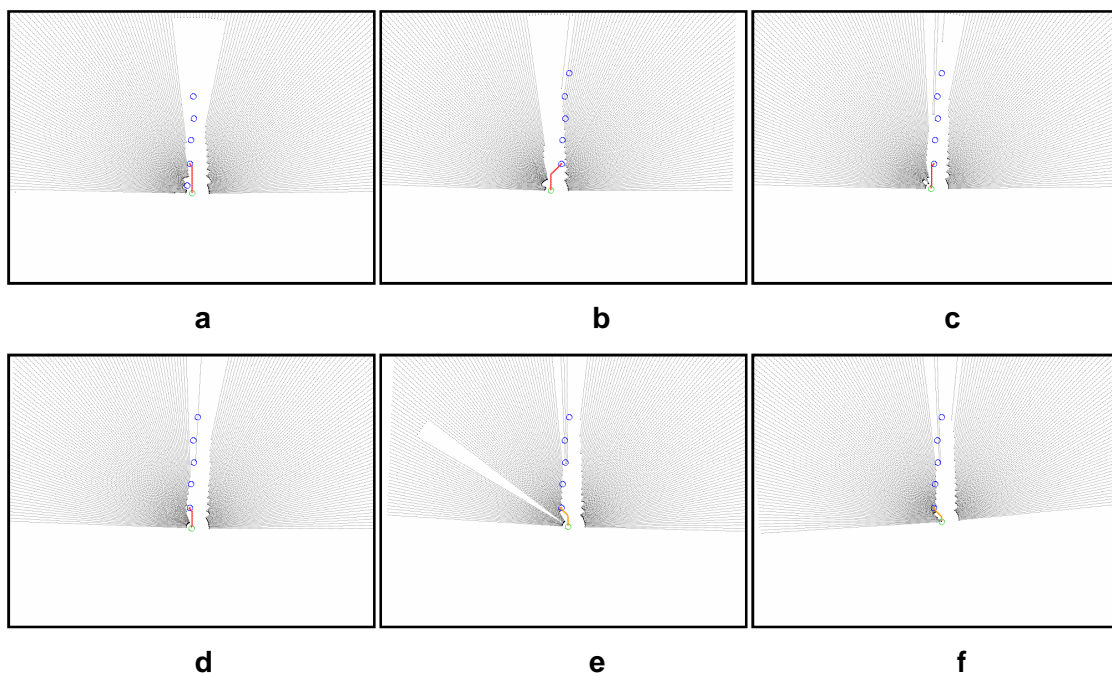
8.2 Pruebas en movimiento

Las pruebas dinámicas comprenden una serie de simulaciones realizadas con el robot en movimiento, donde se realizan las acciones de translación y rotación de forma independiente o conjunta.

El objetivo de estas simulaciones consiste en probar, ante situaciones reales y en tiempo de ejecución, las prestaciones del algoritmo de búsqueda local; además de los factores anteriormente expuestos en las pruebas estáticas, en este caso se suma las dificultades asociadas al movimiento del propio robot.

8.2.1 Simulación 1

La Figura 8-13 muestra una secuencia de mapas de movimiento, donde se realiza la búsqueda del camino local conocidos los puntos a alcanzar o submetas del camino global (círculos azules). En caso de encontrar un camino libre de obstáculos, la ruta a seguir se representa en color rojo; en caso contrario, se obtiene un camino de aproximación al destino más próximo, que se representa en color naranja.



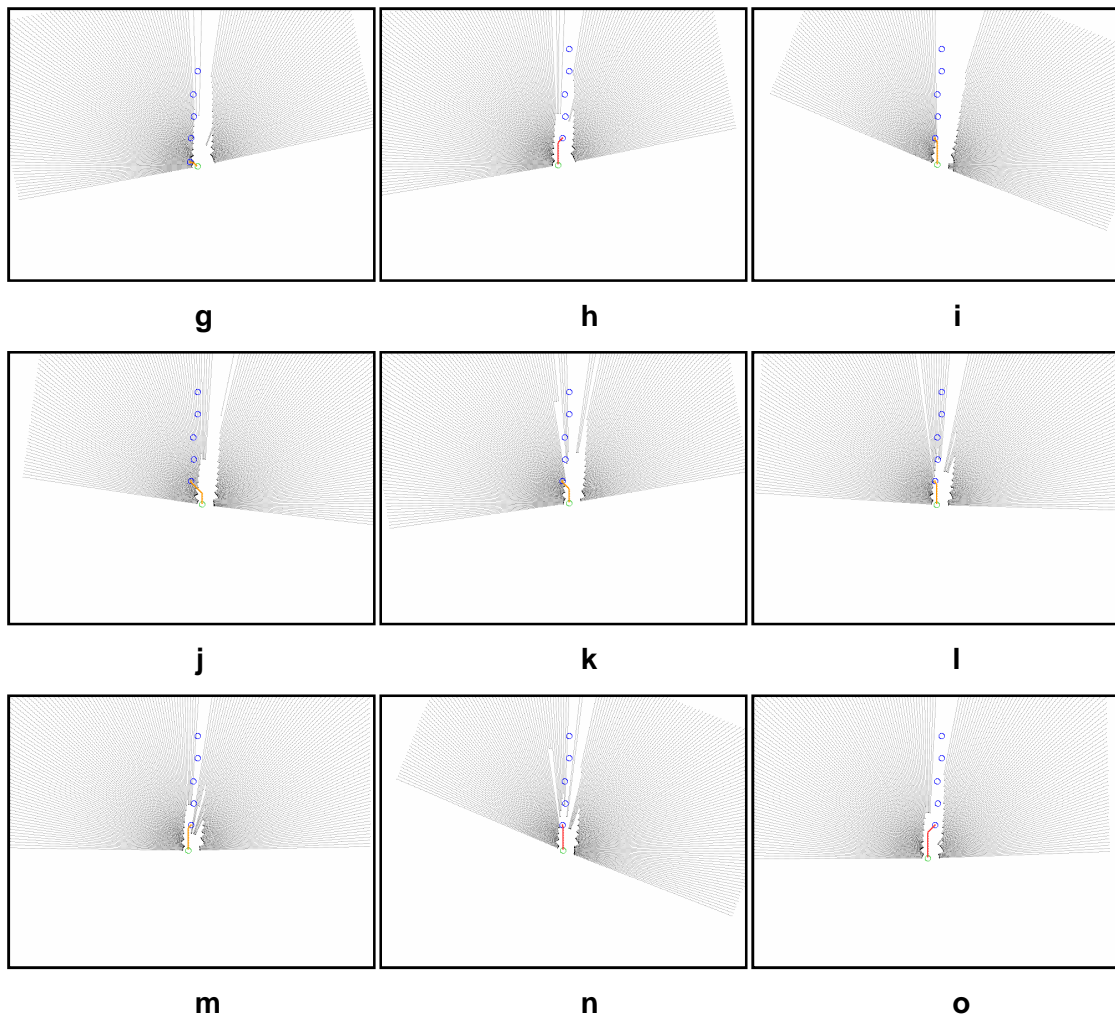


Figura 8-13: búsqueda local – caso dinámico (I)

Alcance del primer objetivo (a,b): la primera submeta del camino global se encuentra próxima a una zona ocupada, de forma que el algoritmo devuelve el camino a la siguiente posición (a); al avanzar, deja atrás esa primera submeta, por ello la elimina de sus objetivos y se centra en la siguiente (b), actualizando la información.

Alcance del segundo objetivo (c,d,e,f,g,h): a medida que avanza, se pierde el camino real (e), lo que lleva a iniciar una rotación a la izquierda (f) para seguir la ruta de aproximación; consigue alcanzar el objetivo (g) y toma el siguiente del camino global (h).

Tercer objetivo (i,j,k,l,m,n,o): se realizan una serie de rotaciones a derecha (i)(n) e izquierda (k) para tratar de obtener el camino a la siguiente submeta; de forma

paralela, se produce el paso de peatones ante el vehículo **(k,l,m,n)**. El algoritmo finalmente encuentra el camino real **(n,o)**.

La Figura 8-14 muestra el mapa de obstáculos obtenido al finalizar la simulación; en rojo, se representan los objetos actualmente medidos, y en azul, los históricos que permanecen en memoria; a diferencia de los mapas de obstáculos estáticos, vistos en el apartado anterior (8.1), en el caso de avance, la posición final de los obstáculos sufre la misma traslación que el vehículo, lo que se aprecia en el desplazamiento de los mismos respecto al origen inicial.

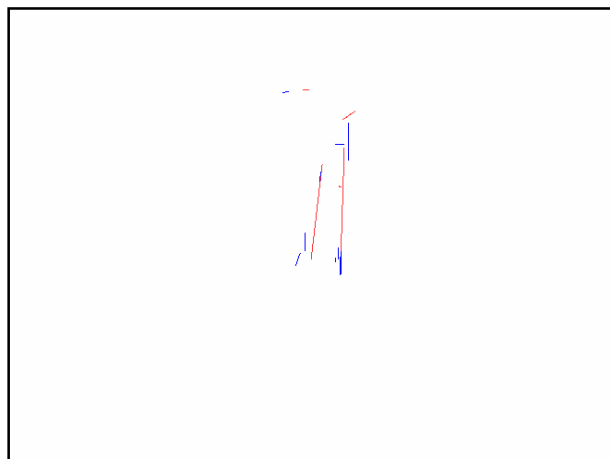


Figura 8-14: mapa de obstáculos al finalizar la simulación

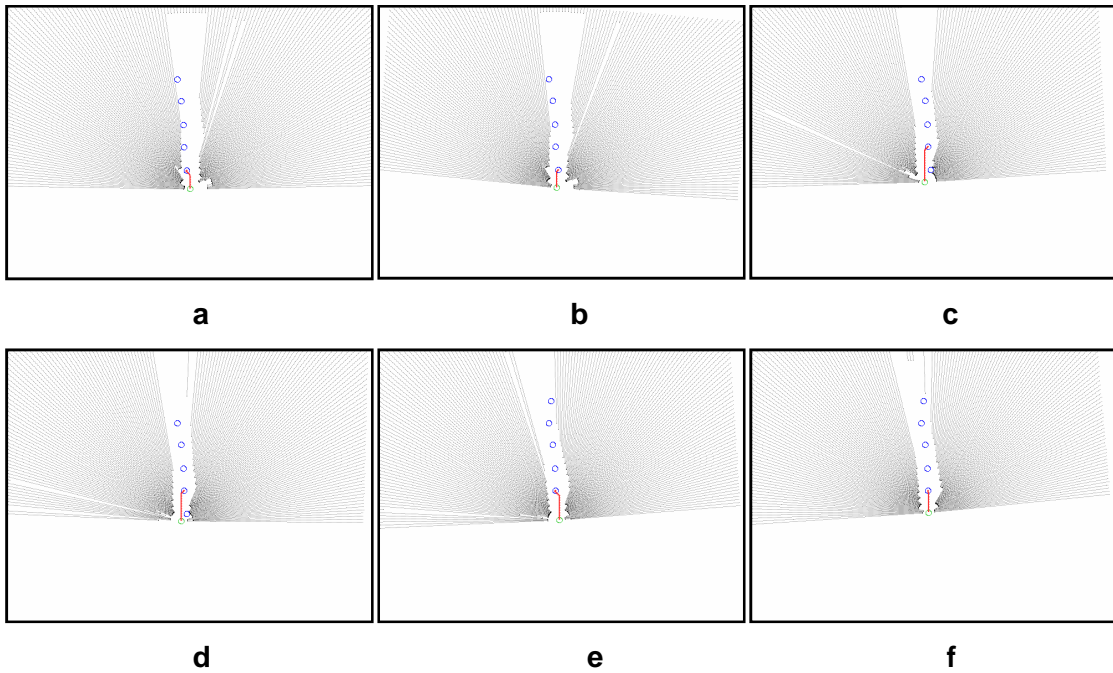
La Figura 8-15 muestra el mapa incremental obtenido al finalizar la simulación, donde se puede apreciar el avance del robot, partiendo de la posición inicial (correspondientes al origen de coordenadas del mapa) y moviéndose en línea recta (siguiendo la vertical o eje Y). Como se comentó en el apartado 5.2, las prestaciones del dispositivo GPS mejoran cuando el vehículo se encuentra en movimiento, obteniéndose la posición de una forma más exacta - con menos errores y desviaciones - que cuando el vehículo se encuentra parado.



Figura 8-15: mapa incremental al finalizar la simulación

8.2.2 Simulación 2

La Figura 8-19 muestra la secuencia de mapas de movimiento, donde los puntos a alcanzar se representan en círculos azules, el camino local encontrado en color rojo, y el camino de aproximación en naranja.



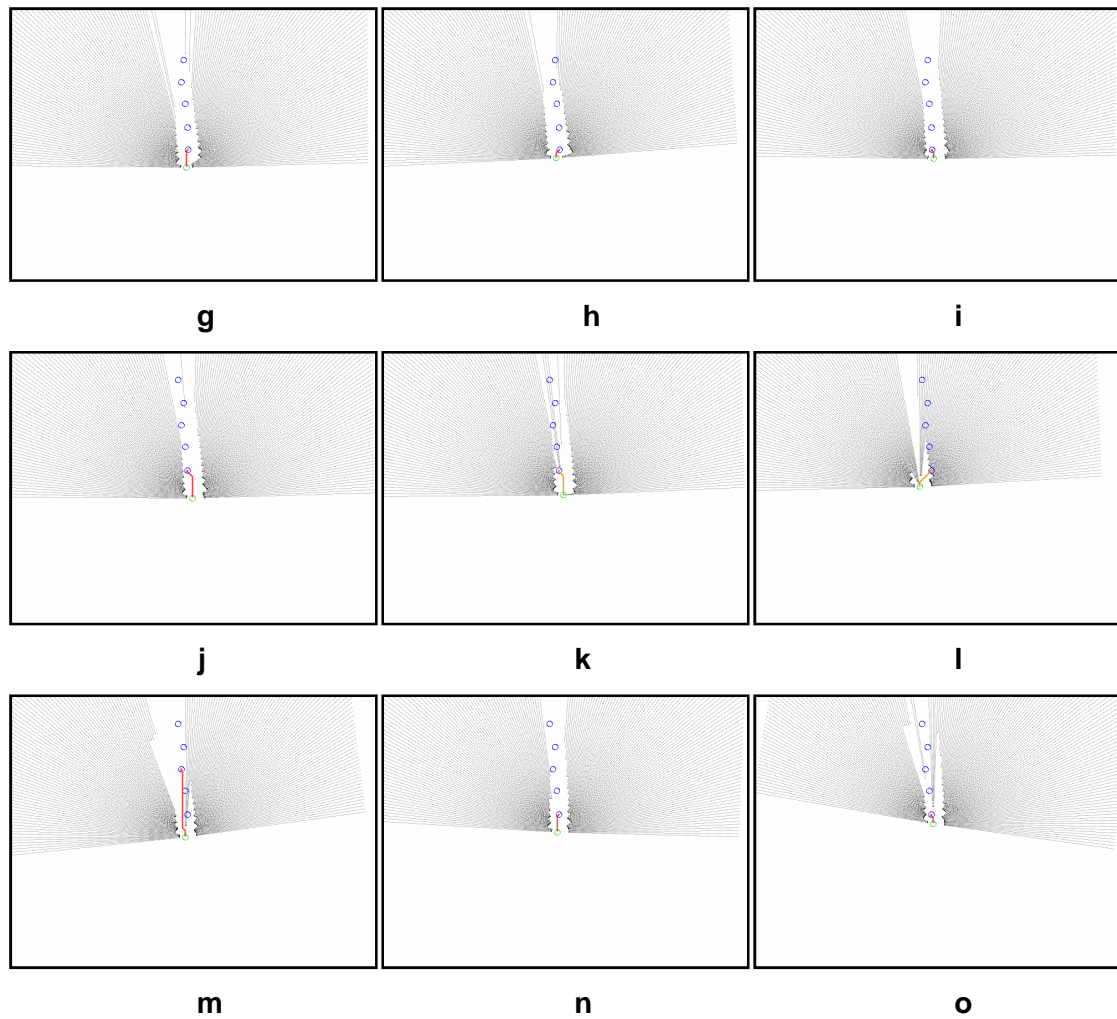


Figura 8-16: búsqueda local – caso dinámico (II)

Alcance del primer objetivo (a,b,c,d,e): el robot intenta alcanzar la primera posición del camino global (a,b); pero al avanzar, dicha posición queda próxima a zona de obstáculos, por lo que toma la siguiente submeta (c,d). Una vez la primera ha quedado atrás, se elimina de los objetivos y se actualiza la información (e).

Alcance del segundo objetivo (f,g,h,i,j): el robot se aproxima al objetivo más cercano, y una vez lo alcanza, toma el siguiente (j).

Tercer objetivo (k,l,m,n,o): la aparición de peatones impide obtener un camino real, por lo que el robot toma la ruta aproximada al objetivo más cercano (k,l) hasta que le posible calcular un camino libre de obstáculos que le dirija a alguna de las

submetas posteriores (**m**). Una vez han desaparecido los obstáculos móviles del campo de visión, se retoma el alcance del primer objetivo (**n,o**).

La Figura 8-17 muestra el mapa de obstáculos obtenido al finalizar la simulación; en rojo, se representan los objetos actualmente medidos, y en azul, los históricos que permanecen en memoria.

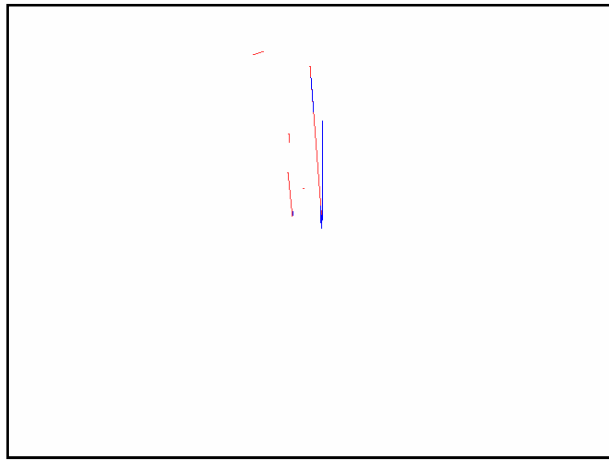


Figura 8-17: mapa de obstáculos al finalizar la simulación

La Figura 8-18 muestra el mapa incremental, donde se puede apreciar el avance del robot en línea recta (siguiendo la vertical o eje Y).

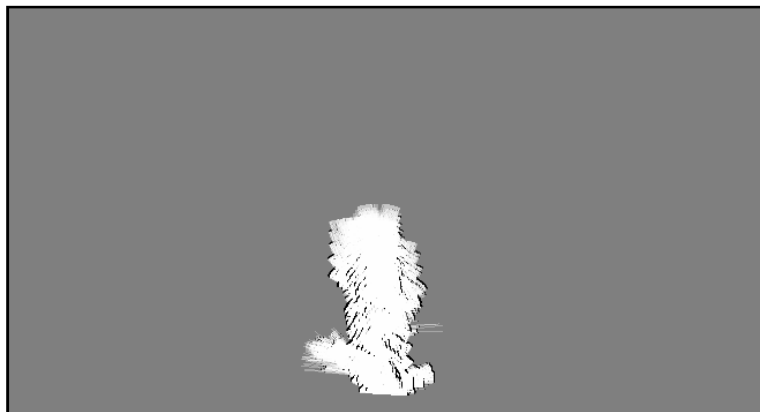
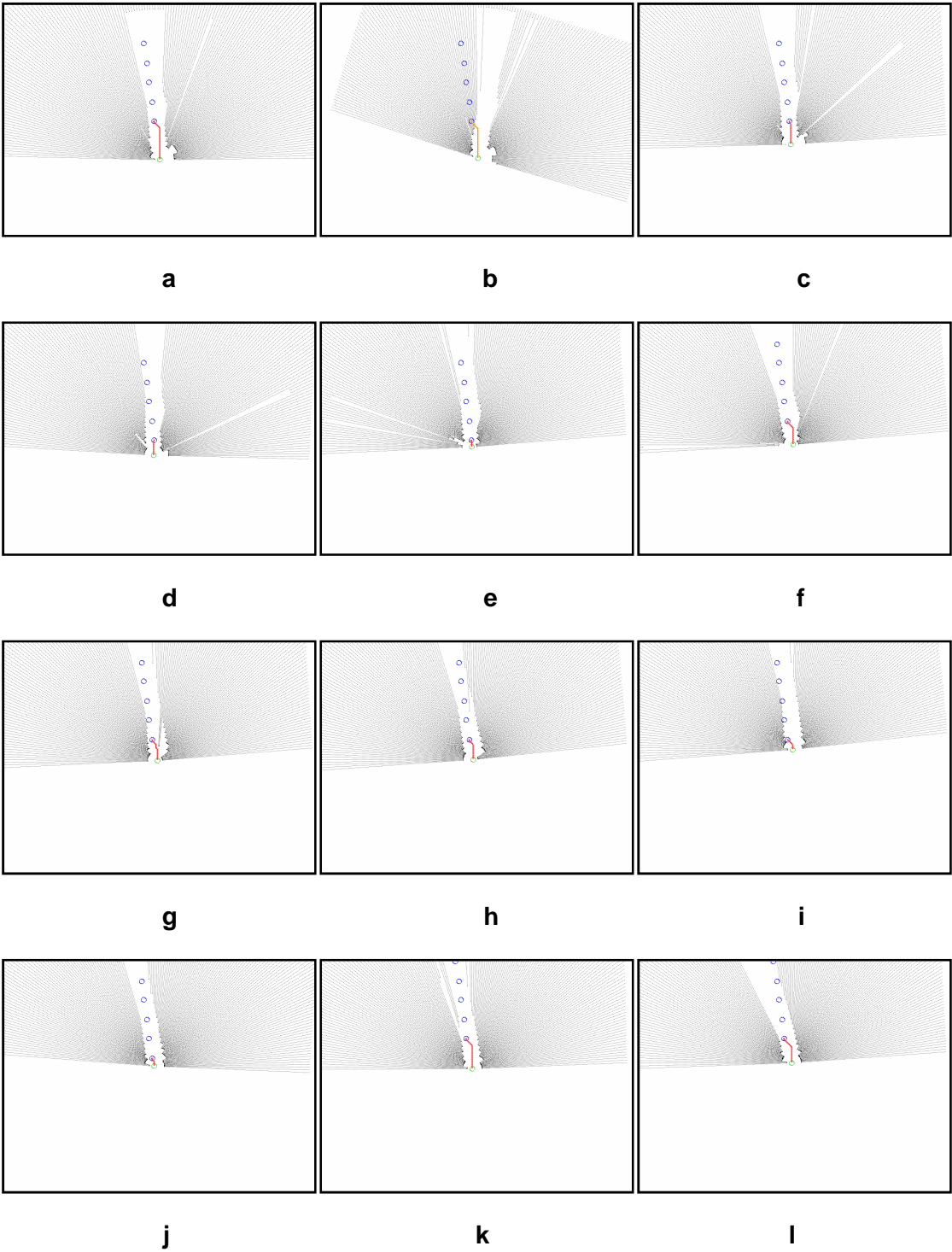


Figura 8-18: mapa incremental al finalizar la simulación

8.2.3 Simulación 3

La Figura 8-19 muestra la secuencia de mapas de movimiento, donde los puntos a alcanzar se representan en círculos azules, el camino local encontrado en color rojo, y el camino de aproximación en naranja.



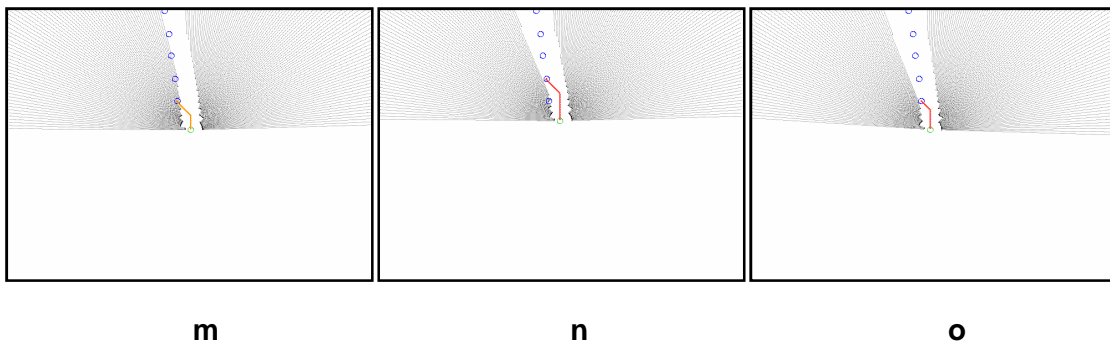


Figura 8-19: búsqueda local – caso dinámico (III)

Alcance del primer objetivo (a,b,c,d,e,f): al inicio del algoritmo, el primer objetivo se encuentra alejado de la posición actual del robot (**a**). A medida que este avanza y gira, es posible la pérdida del camino (**b**), aunque posteriormente lo vuelve a encontrar (**c**). Una vez alcanzada la posición (**e**), se toma la siguiente submeta y se actualiza la información (**f**).

Alcance del segundo objetivo (g,h,i,j,k): el robot avanza al siguiente objetivo, y aunque aparecen obstáculos móviles (**g**), no tiene problemas a la hora de encontrar el camino; una vez alcanza la posición deseada (**j**), toma la siguiente (**k**).

Tercer objetivo (l,m,n,o): se produce la pérdida del camino (**m**) y el cálculo de ruta a una submeta posterior (**n**) para, finalmente, volver a encontrar el camino a la posición más cercana (**o**).

La Figura 8-20 muestra el mapa de obstáculos obtenido al finalizar la simulación; en rojo, se representan los objetos actualmente medidos, y en azul, los históricos que permanecen en memoria.

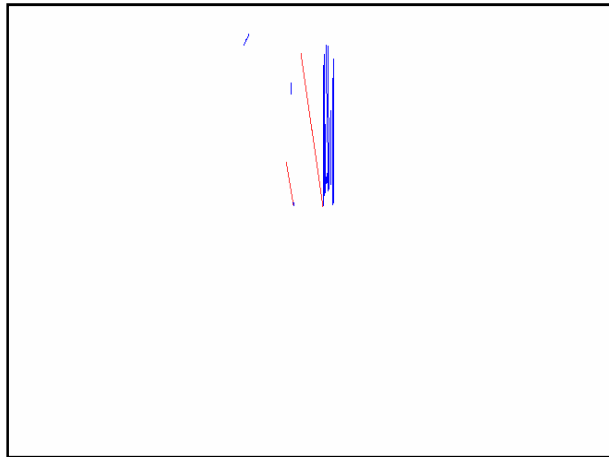


Figura 8-20: mapa de obstáculos al finalizar la simulación

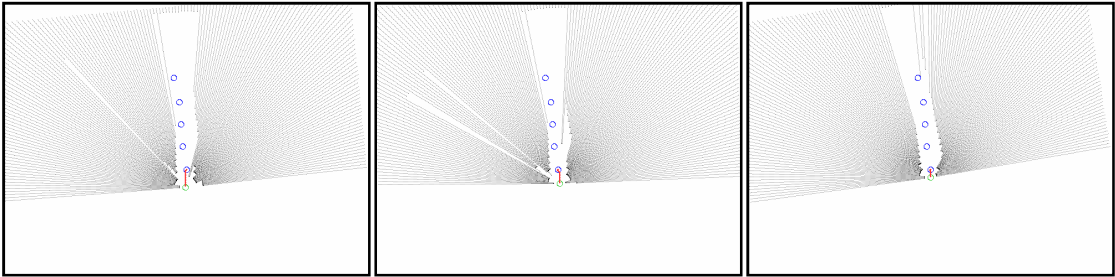
La Figura 8-21 muestra el mapa incremental, donde se puede apreciar el avance del robot en línea recta (siguiendo la vertical o eje Y).



Figura 8-21: mapa incremental al finalizar la simulación

8.2.4 Simulación 4

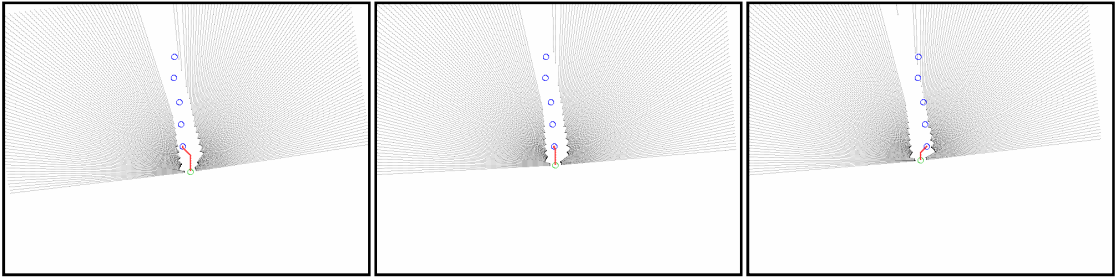
La Figura 8-22 muestra la secuencia de mapas de movimiento, donde los puntos a alcanzar se representan en círculos azules, el camino local encontrado en color rojo, y el camino de aproximación en naranja. En este caso, se trata de una simulación más larga que las anteriores no en cuanto a duración temporal, sino por realizarse un mayor avance espacial y una mayor captura de imágenes.



a

b

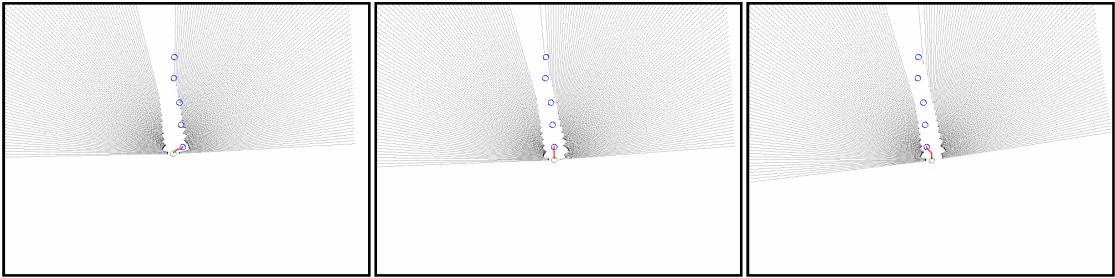
c



d

e

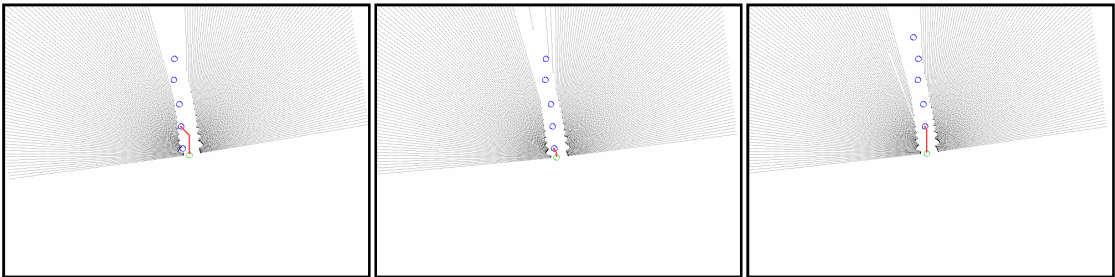
f



g

h

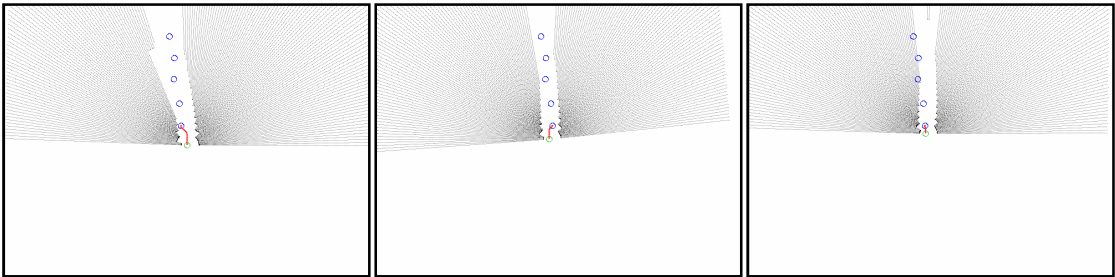
i



j

k

l



m

n

o

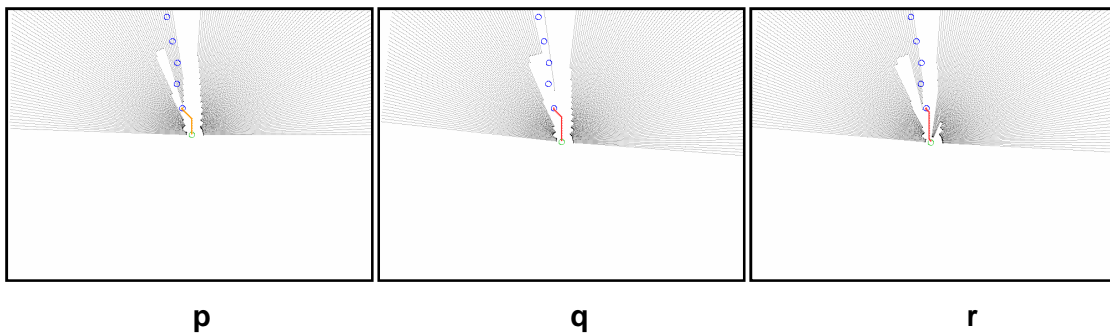


Figura 8-22: búsqueda local – caso dinámico (IV)

Alcance del primer objetivo (a,b,c,d): la presencia de obstáculos móviles (a,b) no impide obtener el camino a la primera submeta, que una vez alcanzada (c) es sustituida por la que va a continuación (d).

Alcance del segundo objetivo (e,f,g,h,i,j,k,l): la existencia de obstáculos fijos próximos a la posición del vehículo no impide obtener el camino al objetivo más cercano (e,f,g), aunque sí lo impide la proximidad de las zonas ocupadas (j), lo que hace necesario buscar una ruta a destinos posteriores.

Alcance del tercer objetivo (m,n,o,p) y seguimiento del cuarto objetivo (q,r).

La Figura 8-23 muestra el mapa de obstáculos obtenido al finalizar la simulación; en rojo, se representan los objetos actualmente medidos, y en azul, los históricos que permanecen en memoria.

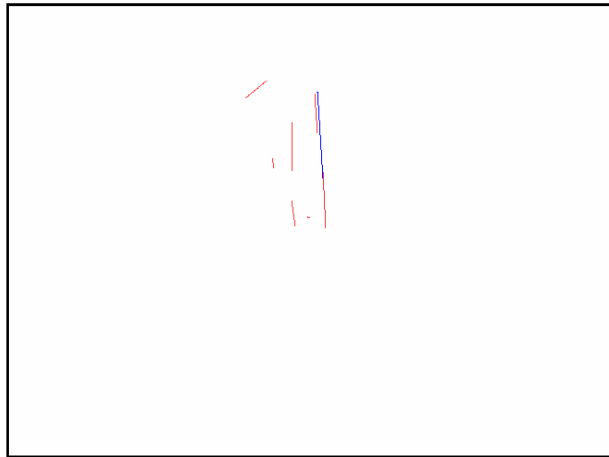


Figura 8-23: mapa de obstáculos al finalizar la simulación

La Figura 8-24 muestra el mapa incremental, donde se puede apreciar el avance del robot en línea recta (siguiendo la vertical o eje Y).

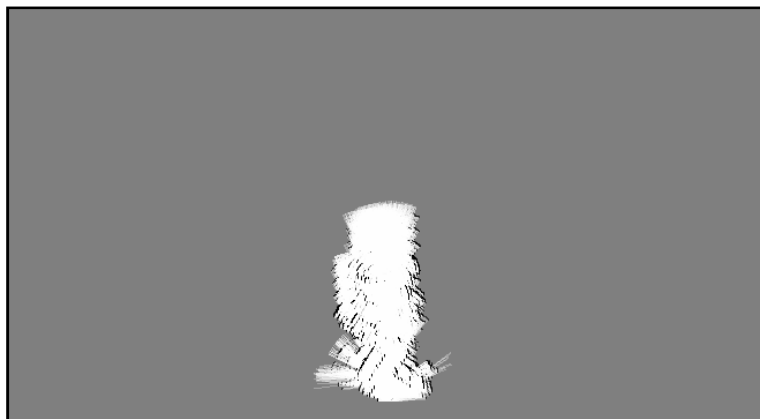
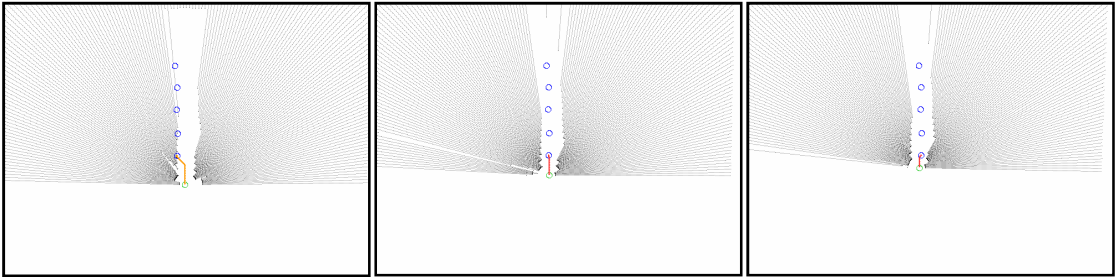


Figura 8-24: mapa incremental al finalizar la simulación

8.2.5 Simulación 5

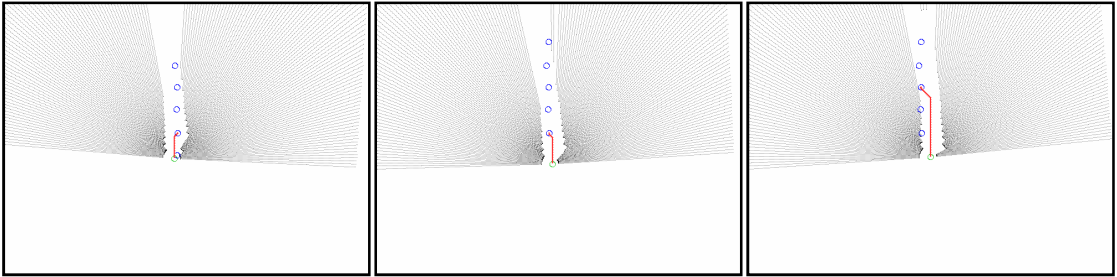
La Figura 8-25 muestra la secuencia de mapas de movimiento, donde los puntos a alcanzar se representan en círculos azules, el camino local encontrado en color rojo, y el camino de aproximación en naranja. Al igual que en el caso anterior, se produce un mayor avance espacial y una mayor captura de imágenes.



a

b

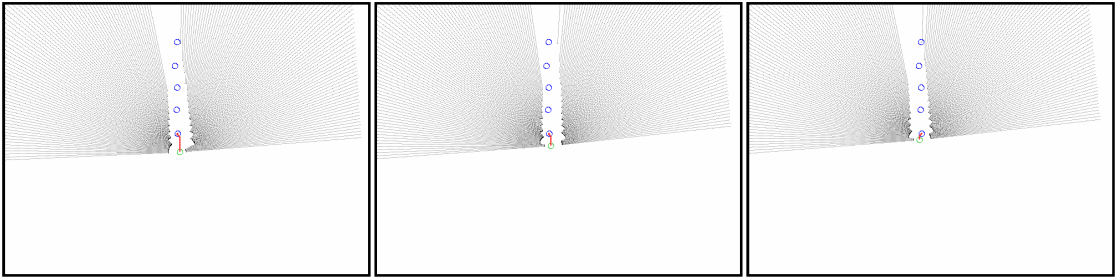
c



d

e

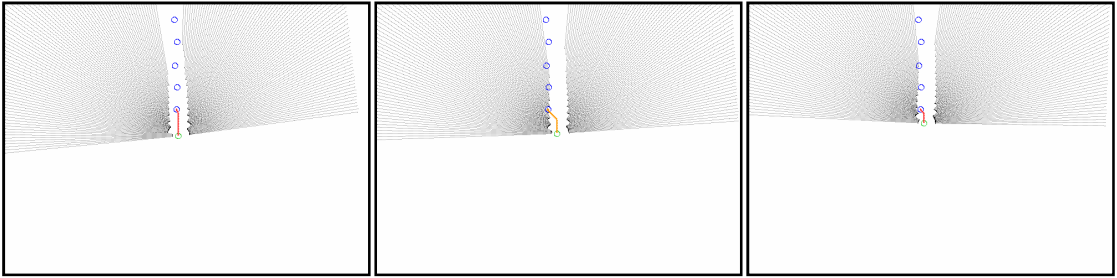
f



g

h

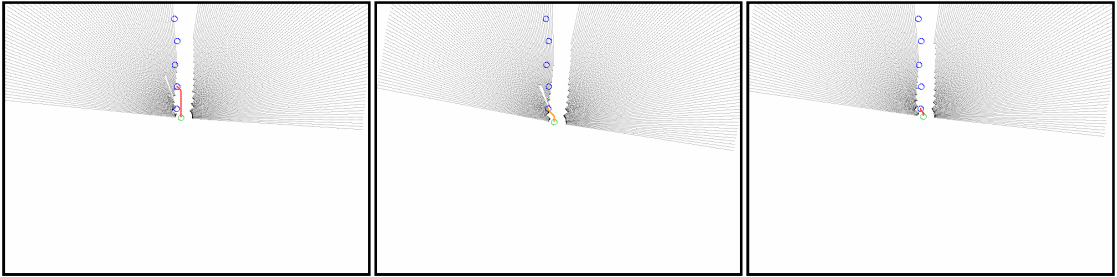
i



j

k

l



m

n

o

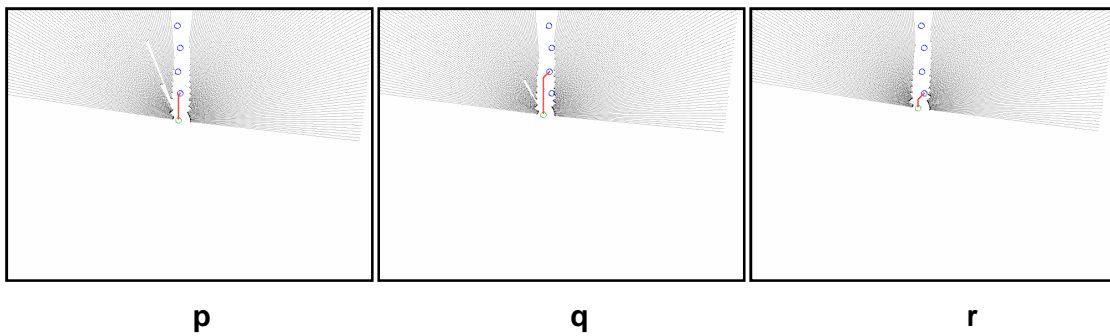


Figura 8-25: búsqueda local – caso dinámico (V)

Alcance del primer objetivo (a,b,c,d): el algoritmo busca el camino al primer objetivo y, una vez alcanzado, toma el siguiente destino (d).

Alcance del segundo objetivo (e,f,g,h,i,j): cuando los objetivos más cercanos se encuentran próximos a zonas ocupadas, el algoritmo devuelve el camino a un destino posterior (f), aunque después vuelva a encontrar la ruta al primer destino (g); una vez alcanzado (i), toma el siguiente (j).

Alcance del tercer objetivo (k,l,m,n,o,p): al igual que antes, si el algoritmo no encuentra camino a la posición más cercana, busca cómo llegar a destinos posteriores (m), aunque en el momento que puede vuelve a la posición próxima (n); una vez alcanzada (o), toma la siguiente (p).

Seguimiento del cuarto objetivo (q,r).

La Figura 8-26 muestra el mapa de obstáculos obtenido al finalizar la simulación; en rojo, se representan los objetos actualmente medidos, y en azul, los históricos que permanecen en memoria.

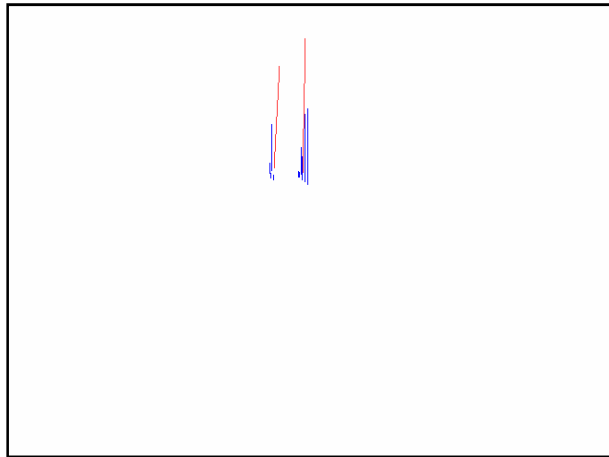


Figura 8-26: mapa de obstáculos al finalizar la simulación

La Figura 8-27 muestra el mapa incremental, donde se puede apreciar el avance del robot en línea recta (siguiendo la vertical o eje Y).



Figura 8-27: mapa incremental al finalizar la simulación

8.3 Comparativa de pruebas

A continuación se presenta la comparativa entre las dos categorías de pruebas realizadas (estáticas y en movimiento), así como los resultados globales obtenidos en cada una de ellas.



Figura 8-28: comparativa de pruebas

La Figura 8-28 muestra las diferencias existentes entre las simulaciones realizadas de forma estática y las realizadas con el vehículo en movimiento.

- En rojo, se representa el porcentaje de casos analizados en los que el algoritmo de búsqueda local no es capaz de encontrar un camino libre de obstáculos, devolviendo en este caso el camino de aproximación a la submeta más próxima.
- En azul, se representa el porcentaje de éxitos al encontrar un camino libre de obstáculos (a alguna de las tres submetas más cercanas a la posición actual).
- En verde, se representan los casos en los que se produce error debido a que el algoritmo encuentra un camino libre que no debería existir, principalmente porque toma como objetivo una submeta que ha quedado atrás (el vehículo la ha rebasado).

Como se puede observar, en el caso estático las probabilidades de encontrar el camino y no encontrarlo son muy parecidas, dándose así mismo situaciones de error; en el caso de movimiento del vehículo, las prestaciones del sistema mejoran al

aumentar la precisión en la localización, llegando a un 80% de casos en los que se encuentra el camino libre de obstáculos, y eliminándose errores en la búsqueda.

A continuación, se exponen los resultados individuales para cada una de las categorías de pruebas (Figura 8-29 y Figura 8-30).

Los motivos que impiden la obtención de un camino libre de obstáculos (representados en tonos rojos) son: la presencia de obstáculos móviles (peatones) que impiden alcanzar el objetivo, y las desviaciones en la posición debido a la deriva del dispositivo GPS; se puede comprobar que esto último es el factor principal de fallos, teniendo mayor relevancia en caso de que el vehículo permanezca parado.

En cuanto a los casos de éxito (representados en tonos azules), cabe destacar que, en la mayor parte de las situaciones analizadas, el algoritmo de búsqueda local es capaz de encontrar el camino a la submeta más cercana, produciéndose la búsqueda respecto a submetas posteriores en un menor número de ocasiones.

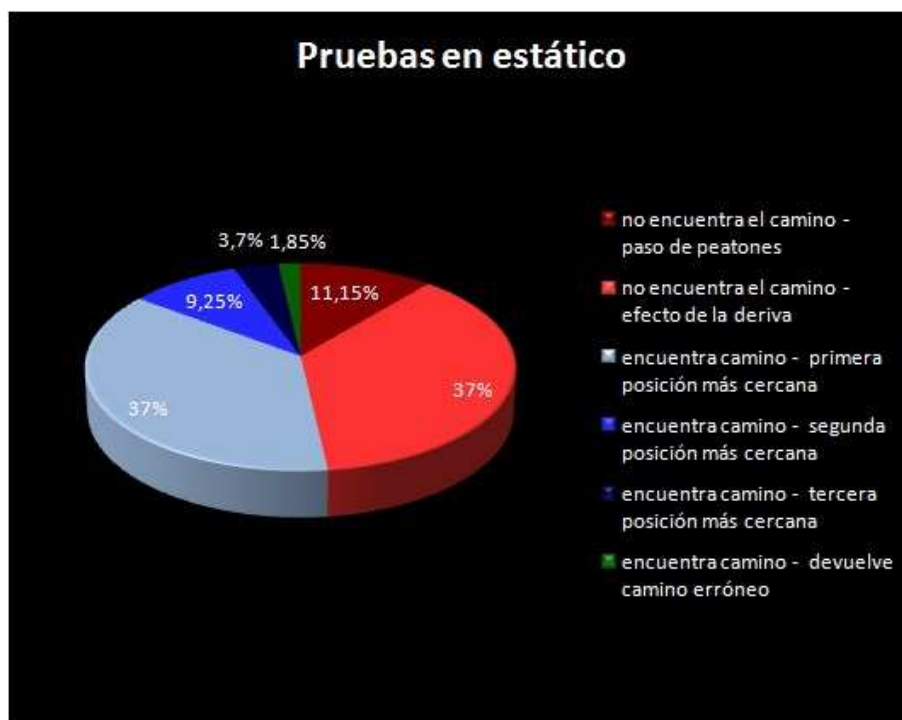


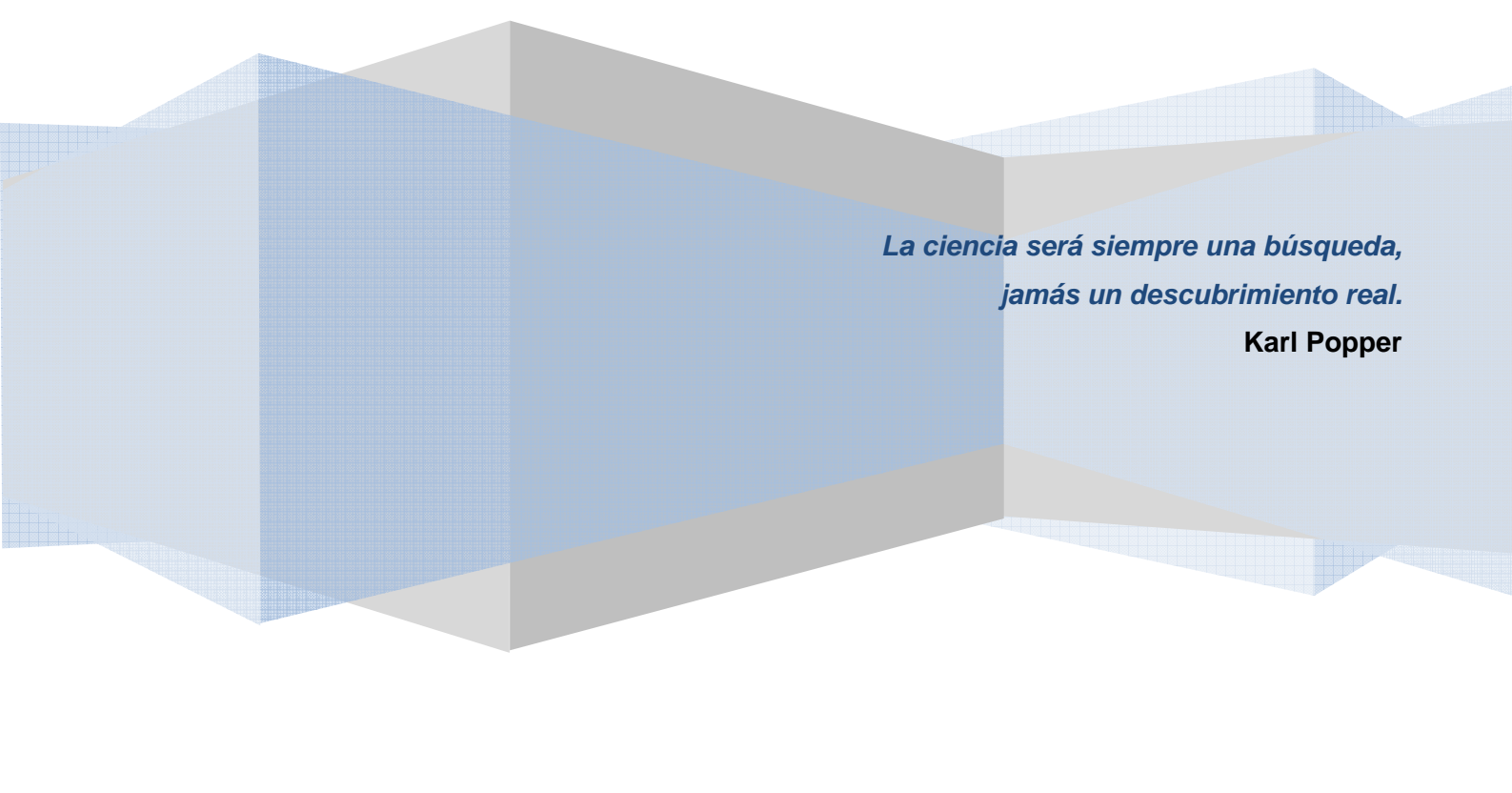
Figura 8-29: resultados de las pruebas en estático



Figura 8-30: resultados de las pruebas en movimiento

CAPÍTULO 9.

CONCLUSIONES Y DESARROLLOS FUTUROS



*La ciencia será siempre una búsqueda,
jamás un descubrimiento real.*

Karl Popper

9. CONCLUSIONES Y DESARROLLOS FUTUROS

A continuación se presentarán las conclusiones generales obtenidas tras el desarrollo del sistema y la realización de las pruebas, así como las futuras líneas de trabajo orientadas a mejorar y ampliar las prestaciones desarrolladas hasta el momento.

9.1 Conclusiones

El presente proyecto se ha centrado en desarrollar un sistema que permita obtener un camino libre de obstáculos, que comunique dos puntos (inicial y final), teniendo en cuenta las circunstancias variables del entorno exterior donde se desarrolla.

Este planteamiento ha derivado en la necesidad de emplear dos tipos de búsqueda: una global, orientada a un entorno más amplio y estático; y una local, que permita el recorrido entre las posiciones del camino global o submetas, y donde se tengan en cuenta las condiciones dinámicas del medio y los imprevistos que se presenten.

De forma previa, se hizo necesario obtener un modelo del entorno donde aplicar los algoritmos de búsqueda, así como un sistema de localización que situara al robot dentro del medio; esto ha llevado a los siguientes desarrollos:

- Implementación de un sistema de obtención de información del medio mediante el empleo del láser.
- Modelización del entorno y tratamiento de la información para crear una serie de mapas de distinto tipo y uso, así como un esquema de los obstáculos existentes.
- Implementación de un sistema de localización del vehículo vía GPS.

El empleo del dispositivo láser ha resultado satisfactorio en el presente proyecto, devolviendo información estable y fiable del medio; de esta forma, los mapas creados demuestran ser adecuados para aplicar las técnicas de búsqueda de caminos. Esto lleva a plantear el empleo del láser como elemento fundamental para la aplicación del sistema implementado en el presente proyecto.

Sin embargo, la localización ha presentado mayores problemas, debido a la inexactitud e inestabilidad de la información procedente del GPS. Esta situación ha generado dificultades a lo largo de todo el proceso, impidiendo obtener resultados correctos en algunas ocasiones, principalmente aquellas en las que el vehículo se hallaba parado; sin embargo, se ha podido comprobar que en movimiento, el porcentaje de éxito en la búsqueda del camino experimentaba un aumento considerable.

Por lo tanto, no es preciso descartar completamente el uso del dispositivo GPS, aunque sí sería conveniente tener en cuenta su posible falta de precisión, mediante la comparación entre medidas consecutivas para descartar errores, y mediante el apoyo de los datos proporcionados por otros mecanismos de localización (no desarrollados en el presente proyecto).

En cuanto al algoritmo de búsqueda, tanto global como local, las prestaciones que ofrece satisfacen las necesidades del sistema que se quería implementar, permitiendo alcanzar los objetivos deseados. Una de las mejoras aplicadas al mismo, consistente en la búsqueda de submetas posteriores en caso de no encontrar el camino a la más cercana, ha demostrado su eficacia y el aumento del porcentaje de éxito. Por otra parte, el hecho de que, al no encontrar el camino, se obtenga una aproximación a la submeta más cercana, representa otra mejora de interés, ya que permite al vehículo obtener en todo momento la información necesaria para desplazarse, permitiéndole salir de situaciones conflictivas y buscando estabilizar la posición.

9.2 Desarrollos futuros

Entre las mejoras relacionadas con el presente desarrollo, cabría destacar:

- Mejorar el sistema de localización del vehículo, empleando técnicas de SLAM (en lugar del dispositivo GPS, o como apoyo a los datos proporcionados por éste); una idea interesante sería el empleo de cámaras de visión y técnicas de tratamiento de imágenes.
- Relacionado con lo anterior, y empleando técnicas de SLAM, prescindir del mapa global y centrar la búsqueda a una serie de objetivos locales.
- Otra opción para mejorar el sistema de localización del vehículo, sería la de emplear información odométrica para realizar un ajuste de la posición proporcionada por el GPS.
- Aplicar algún tipo de filtrado a la información procedente del láser (por ejemplo, mediante Bayes), para refinar los resultados obtenidos entre medidas consecutivas y eliminar posibles errores.
- Mejorar la visualización de los mapas y la interfaz con el usuario, permitiéndole la elección de los puntos inicial y final del camino de forma dinámica sobre el mapa global del entorno.
- Dotar al sistema de un sistema de aprendizaje que le permita no sólo el avance, sino también el retroceso del vehículo.
- En relación al almacenamiento de obstáculos, ampliar el tratamiento que reciben los mismos, mediante separación (de un mismo obstáculo en dos obstáculos distintos) y unión (realizando una combinación a partir de aristas comunes para formar un solo obstáculo).

Otro tipo de desarrollos relacionados con aspectos no contemplados en el presente proyecto, podrían ser:

- Implementar un sistema de seguimiento de ruta (una vez conocidos los puntos del camino) y de evitación de obstáculos.
- Emplear un segundo láser que, barriendo una superficie vertical, permitiese combinar información relativa a la altura con la ya obtenida horizontalmente; de esta forma, los obstáculos podrían modelarse como estructuras tridimensionales, además de permitir la detección de los elementos de reducidas dimensiones más próximos al suelo.
- Clasificar los obstáculos según su tipo, forma, tamaño e influencia en la obtención del camino, así como realizar un seguimiento de los mismos mientras se encuentren dentro del área de visión del láser; en este aspecto, sería interesante tener en cuenta la correlación entre distintas medidas para deducir si se trata del mismo objeto en movimiento, y calcular su velocidad y dirección para que el robot reaccione de forma adecuada.

BIBLIOGRAFÍA

*Si no entiendes algo puedes hacer que
signifique cualquier cosa.*

Chuck Palahniuk

Bibliografía

[AAP] Australia Air Power - Australia's Independent Defence Think Tank [En línea]. - 2009. - <http://www.ausairpower.net/TE-GPS-Guided-Weps.html>.

[Cañ07] José María Cañas Plaza // Introducción a la Robótica. - [s.l.] : Universidad Rey Juan Carlos, Madrid, 2007.

[Cas02] Cristina Castejón Sisamón // Modelado de zonas cruzables en entornos exteriores para robots móviles. - [s.l.] : Universidad Carlos III, Madrid, 2002.

[DEI] Departamento de Electrónica e Informática [En línea] // Universidad Católica Nuestra Señora de la Asunción - Paraguay. - 2009. - <http://www.dei.uc.edu.py/tai99/robotica/sistemas.htm>.

[Gal07] Cipriano Galindo Andrades // Modelado del entorno en robótica móvil. - Zaragoza : II Congreso Español de Informática (CEDI), Septiembre de 2007.

[Gar09-lidar] F. García, F. Jiménez, J.E. Naranjo, J.M. Armingol, A. de la Escalera, F.Aparício // Analysis of LIDAR sensors for new ADAS applications. Usability in moving obstacles detection. - [s.l.] : Universidad Carlos III de Madrid, Universidad Politécnica de Madrid, 2009.

[Gar09-obs] Fernando García Fernández // Detección y clasificación de obstáculos en entornos viarios mediante radar láser. - [s.l.] : Universidad Carlos III, Madrid, Mayo de 2009.

[Gar09-veh] Fernando Garcia, Pietro Cerri, Alberto Broggi, Jose Maria Armingol, Arturo de la Escalera // Vehicle Detection Based on Laser Radar. - [s.l.] : Universidad Carlos III de Madrid, Universit degli Studi di Parma, 2009.

[Hid09] Juan Manuel Hidalgo Fiestas // iCab. Automatización de un vehículo eléctrico. Mecánica y software. - [s.l.] : Universidad Carlos III, Madrid, Julio de 2009.

[LIDAR] LIDAR - scanner [En línea]. - <http://en.wikipedia.org/wiki/File:LIDAR-scanned-SICK-LMS-animation.gif>.

[Lop08-heu] Abraham Sánchez López // Técnicas heurísticas en planificación de movimientos. - [s.l.] : FCC/BUAP.

[Lop08-mov] Abraham Sánchez López // Planificación de movimientos. - [s.l.] : FCC/BUAP, 2008.

[Lop08-rea] Abraham Sánchez López // Planificación reactiva de movimientos. - 2008.

[Lop08-sen] Abraham Sánchez López // Planificación de movimientos basada en sensores. - 2008.

[Muñ95] Victor Fernando Muñoz Martínez // Planificación de Trayectorias para Robots Móviles. - [s.l.] : Universidad de Málaga, 5 de Julio de 1995.

[Nil09] Nils Ryden [En línea]. - 2009. - http://www.masw.com/files/Local-GlobalMinima_small_-1.JPG.

[OYP] Ordenadores y Portátiles [En línea]. - 2009. - <http://www.ordenadores-y-portatiles.com/gps.html>.

[Por09] Pedro Portalatín Ginés // iCab. Automatización de un vehículo eléctrico. Diseño hardware. - [s.l.] : Universidad Carlos III, Madrid, Julio de 2009.

[Pue07] Paloma de la Puente Yusty // Sistema de navegación de un robot móvil. - [s.l.] : Universidad Politécnica de Madrid, Noviembre de 2007.

[Rao98] N.S.V Rao, S.S Iyengar, G. deSaussure // The Visit Problem: Visibility Graph-Based Solution. - 1998.

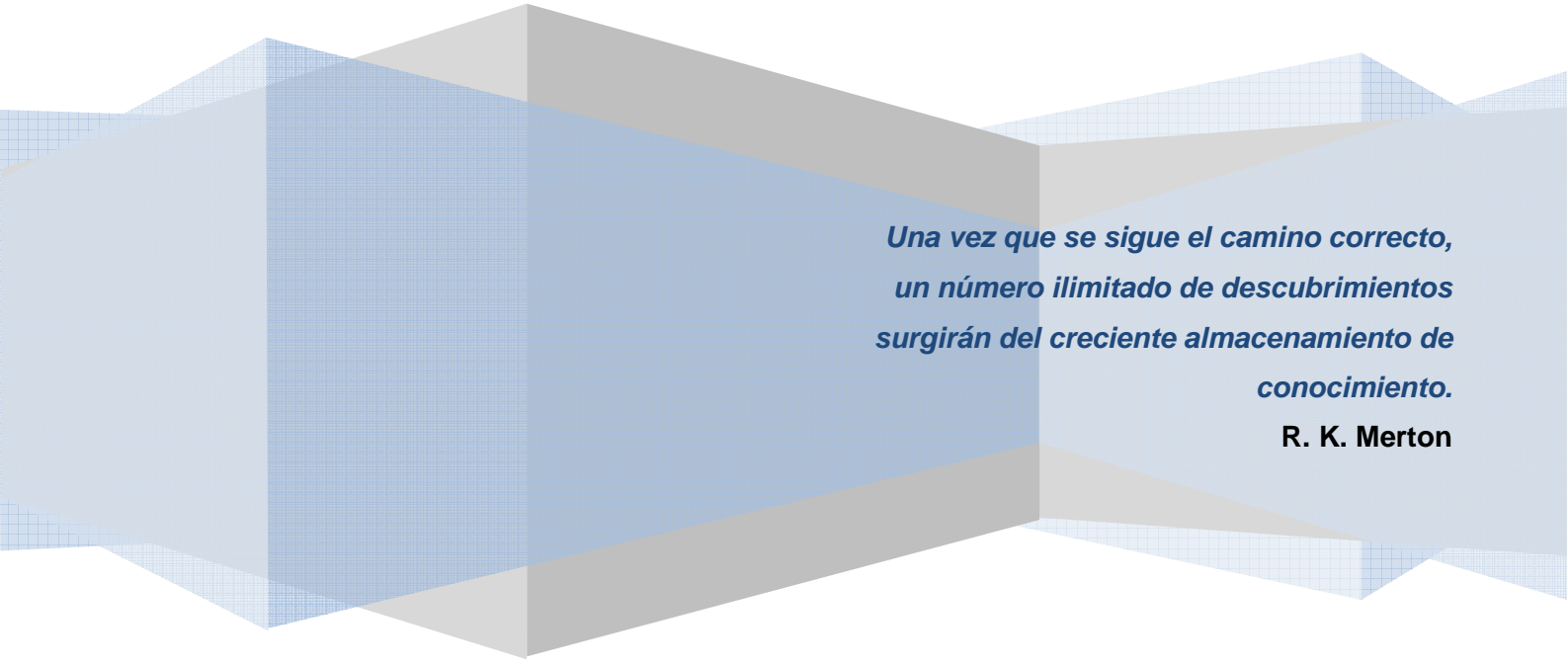
[San04] Erik Adolfo de los Santos de la Rosa // Heurística para la generación de configuraciones en pasajes estrechos aplicada al problema de los clavos. - [s.l.] : Universidad de las Américas Puebla, 17 de Mayo de 2004.

[SICK] SICK - Hojas de características del modelo LMS 291-S05 [En línea]. - <http://www.sick.es/es/es.html>.

[Suc09] L. Enrique Sucar // Introducción a la Robótica. - [s.l.] : Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), 2009.

[Thr05] Sebastian Thrun, Wolfram Burgard, Dieter Fox // Probabilistic Robotics. - [s.l.] : The MIT Press, 2005.

ANEXOS



*Una vez que se sigue el camino correcto,
un número ilimitado de descubrimientos
surgirán del creciente almacenamiento de
conocimiento.*

R. K. Merton

Anexos

Transformaciones de coordenadas

El sistema de referencia asociado a cada uno de los mapas implementados en el proyecto, varía dependiendo del tipo y naturaleza de los mismos; por ello, se hace preciso una breve explicación sobre los sistemas de coordenadas empleados.

El empleo del dispositivo GPS para la localización permite obtener valores dentro de una escala que se puede considerar como absoluta e independiente del sistema de representación. Esta información está formada por la latitud, la longitud y la orientación de la medida. Dicha orientación representa el ángulo de apuntamiento, esto es, el ángulo que forma el eje central del dispositivo GPS respecto al Norte geográfico. A partir de ahora, las consideraremos como medidas de ámbito global, siendo el Norte la orientación fijada como invariable. La Figura 11-1 representa las coordenadas geográficas en las que se sitúa el entorno del campus de la Universidad Carlos III.

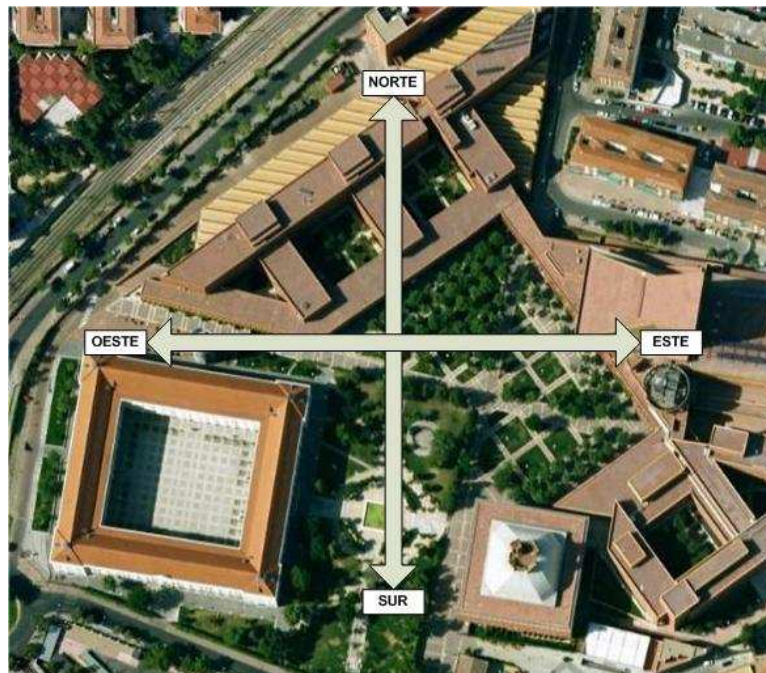


Figura 11-1: ejes geográficos sobre mapa del campus

La representación de los mapas se realiza mediante sistemas bidimensionales con ejes cartesianos X e Y, cuya disposición dependerá de la orientación elegida. Se trata de planos relacionados con la referencia global proporcionada por los valores GPS.

En la construcción del mapa global del entorno, se parte del sistema cartesiano para obtener los valores GPS correspondientes. El centro de coordenadas se sitúa en el punto medio de la imagen que lo representa. Las coordenadas X e Y de las posiciones inicial y final, así como las correspondientes a puntos del camino, son transformadas en duplas (latitud, longitud), proporcionando valores no dependientes de la representación y, por tanto, pudiendo emplearse posteriormente tal cual. La orientación del mapa será la global, lo que quiere decir que el eje vertical (eje Y) coincidirá con el Norte geográfico (Figura 11-2).



Figura 11-2: ejes cartesianos sobre mapa del campus

Los mapas locales que representan únicamente el entorno actual del robot se encuentran referenciados a la posición del mismo, y la orientación del mapa (eje Y) se corresponderá con la orientación del robot en cada momento (Figura 11-3).



Figura 11-3: ejes cartesianos con origen la posición del robot

En el caso del mapa que representa el robot en movimiento, el tratamiento de las coordenadas se complica. Para facilitar los cálculos, el entorno (mapa completo) se fija a la posición de partida del algoritmo, que se considera como la referencia absoluta del mapa; dentro de él, se modela el sistema asociado al robot a medida que se mueve, que será considerado como referencia relativa (en comparación con el anterior). La orientación del mapa completo (eje Y del mapa) depende de la orientación inicial respecto al Norte; la orientación del robot depende de la que tenga en cada momento respecto al Norte.

La Figura 11-4 representa los distintos sistemas de coordenadas empleados: en azul claro, la referencia absoluta (vinculada a la posición inicial del robot); en azul oscuro, las referencias relativas al movimiento del vehículo (donde se aprecian los cambios debidos a la traslación y la rotación del origen de coordenadas).



Figura 11-4: referencia global y referencias locales

